



Centro Universitário de Brasília – UniCEUB

Faculdade de Tecnologia e Ciências Sociais Aplicadas – FATECS

Projeto Final

Remasterização de uma distribuição GNU/Linux visando
uma solução automatizada para servidores de e-mail

Thiago de Almeida Milhomem

RA: 2046610/3

Monografia de Conclusão do Curso de Engenharia de Computação

Professor orientador: M.Sc. Antônio José Gonçalves Pinto

Brasília – DF, junho de 2009

Thiago de Almeida Milhomem

**Remasterização de uma distribuição GNU/Linux visando
uma solução automatizada para servidores de e-mail**

Trabalho de conclusão do curso de
Engenharia de Computação da
Faculdade de Tecnologia e Ciências
Sociais Aplicadas, para obtenção do
título de Bacharelado em Engenharia
de Computação pelo Centro
Universitário de Brasília – UniCEUB.

Professor orientador: M.Sc. Antonio
José Gonçalves Pinto

Brasília – DF, junho de 2009

AGRADECIMENTOS

Gostaria de agradecer à minha família pelo apoio e paciência durante o período de realização desse projeto.

Ao meu amigo e chefe, Alfredo Gomes da Silva Júnior, pelo compartilhamento de idéias e apoio.

À minha namorada, Dayanna de Menezes Martins, pela atenção, compreensão e carinho.

RESUMO

Este projeto apresenta uma solução para servidores de correio eletrônico, com o foco na facilidade na administração, que possa ser instalada e configurada de forma simples e rápida. Isso é feito através da remasterização da imagem de instalação de um sistema GNU/Linux, que possui a finalidade de automatizar todo o processo de instalação e configuração do servidor. São abordados diversos temas relacionados ao serviço de e-mail, utilizados na implementação do servidor. Também é discutido todo o processo de automatização da instalação e configuração do servidor. A segurança da solução é garantida através de criptografia e ferramentas que auxiliam no combate aos spams. Toda a solução é composta de softwares livres, o que descarta a necessidade de gastos com licenças de software.

Palavras-chaves: e-mail, segurança, automatização

ABSTRACT

This project provides a solution for mail servers with a focus on easiness of administration, which can be installed and configured in a simple and fast form. This is done through the remastering of installation image of a GNU/Linux system, which has the purpose of automating the entire process of installing and configuring the server. Diverse subjects related to the mail service are aborted, that had been used in the implementation of the server. It also be discussed the process of automating the installation and configuration of the server. The security of the solution is ensured by encryption and tools that help to combat the spams. The whole solution is implemented with free software, which discards the necessity of expenses with software licenses.

Keywords: mail, security, automatization.

LISTAS DE FIGURAS

FIGURA 1: “MODELO DE VIRTUALIZAÇÃO UTILIZADO NO PROJETO”	16
FIGURA 2: “FUNCIONAMENTO DO SISTEMA DE E-MAIL”	19
FIGURA 3: “ENVIO DE E-MAIL VIA LINHA DE COMANDO”	21
FIGURA 4: “FUNCIONAMENTO DA ENTREGA DE UMA MENSAGEM”	23
FIGURA 5: “SESSÃO POP3 VIA LINHA DE COMANDO”	25
FIGURA 6: “FUNCIONAMENTO DO IMAP”	26
FIGURA 7: “COMANDOS IMAP NO ESTADO AUTENTICADO”	29
FIGURA 8: “COMANDOS IMAP NO ESTADO DE SELEÇÃO”	29
FIGURA 9: “CAIXA POSTAL NO FORMATO MBOX”	30
FIGURA 10: “ESTRUTURA DE DIRETÓRIOS DO FORMATO MAILDIR”	31
FIGURA 11: “ORGANIZAÇÃO DAS MENSAGENS NO FORMATO MAILDIR”	32
FIGURA 12: “CENÁRIO PROPOSTO PARA A IMPLEMENTAÇÃO”	40
FIGURA 13: “OPÇÃO DE UTILIZAR UMA IMAGEM ISO 9660 NO SOFTWARE VMWARE SERVER.”	42
FIGURA 14: “TELA INICIAL DE INSTALAÇÃO DO MANDRIVA LINUX.”	43
FIGURA 15: “MENU DE PARTICIONAMENTO DO INSTALADOR DO MANDRIVA LINUX.”	45
FIGURA 16: “TELA DE SELEÇÃO DE PACOTES DO INSTALADOR DO MANDRIVA LINUX.”	46
FIGURA 17: “CRIAÇÃO DA BASE DE DADOS POSTFIX NO BANCO DE DADOS.”	48
FIGURA 18: “ARQUIVOS DO SOFTWARE POSTFIX ADMIN.”	48
FIGURA 19: “CRIAÇÃO DAS TABELAS PELO POSTFIX ADMIN.”	51
FIGURA 20: “ESTRUTURA DE TABELAS CRIADA PELO POSTFIX ADMIN.”	52
FIGURA 21: “INTERFACE DE ADMINISTRAÇÃO DO POSTFIX ADMIN.”	53
FIGURA 22: “ARQUIVOS DE CONFIGURAÇÃO DO POSTFIX.”	54
FIGURA 23: “PARÂMETROS DO ARQUIVO VIRTUAL-MAILBOX-MAPS.CF.”	56
FIGURA 24: “CRIAÇÃO DA BASE DE DADOS ROUNDCEMAIL.”	59
FIGURA 25: “ARQUIVOS DO SOFTWARE ROUNDCEMAIL.”	60
FIGURA 26: “TELA DO LOGIN DO WEBMAIL ROUNDCEMAIL.”	63
FIGURA 27: “REGISTRO DE MTAS AUTORIZADOS PELO DOMÍNIO UOL.COM.BR.”	66
FIGURA 28: “LÓGICA IMPLEMENTADA NOS MTAS COM SUPORTE AO GREYLISTING.”	68
FIGURA 29: “CRIAÇÃO DA BASE DE DADOS DSPAM.”	71
FIGURA 30: “INTERFACE WEB DO DSPAM.”	77
FIGURA 31: “CERTIFICADOS UTILIZADOS PARA OS PROTOCOLOS HTTPS, IMAPS E POPS”	79
FIGURA 32: “PROCESSO DE MONTAGEM DA IMAGEM DE INSTALAÇÃO.”	82
FIGURA 33: “DIRETÓRIO I586 SITUADO NA IMAGEM DE INSTALAÇÃO DO SISTEMA.”	83
FIGURA 34: “CONTEÚDO DO DIRETÓRIO STAGE2.”	84
FIGURA 35: “ESTRUTURA DO DIRETÓRIO SQUASHFS-ROOT.”	84
FIGURA 36: “ESTRUTURA DE PASTAS DO DIRETÓRIO CONFIGURADOR.”	86
FIGURA 37: “CONTEÚDO DO DIRETÓRIO BIN.”	88
FIGURA 38: “TELA INICIAL DO SCRIPT BENVINDO.SH.”	95
FIGURA 39: “TOPOLOGIA UTILIZADA NOS TESTES.”	98

FIGURA 40: “DOMÍNIOS CRIADOS PARA TESTES NA FERRAMENTA POSTFIX ADMIN.”	100
FIGURA 41: “ENVIO DE UMA MENSAGEM DE TESTE ATRAVÉS DE UM CLIENTE DE E-MAIL.”	100
FIGURA 42: “MENSAGEM VISUALIZADA NO WEBMAIL DO PROJETO.”	101
FIGURA 43: “TESTE DE ENVIO DE MENSAGEM PARA UM DOMÍNIO NA INTERNET.”	102
FIGURA 44: “TESTE DE RECEBIMENTO DE MENSAGEM DE UM DOMÍNIO NA INTERNET.”	102
FIGURA 45: “TOPOLOGIA DE TESTES COM A FERRAMENTA WIRESHARK.”	103
FIGURA 46: “COMUNICAÇÃO IMAP EM TEXTO CLARO.”	104
FIGURA 47: “COMUNICAÇÃO IMAP CRIPTOGRAFADA.”	105
FIGURA 48: “ERRO APRESENTADO PELO NAVEGADOR DE INTERNET.”	105
FIGURA 49: “COMUNICAÇÃO HTTP CRIPTOGRAFADA.”	106
FIGURA 50: “ERRO NA TENTATIVA DE ENVIO DE MENSAGEM SEM A AUTENTICAÇÃO HABILITADA.”	107
FIGURA 51: “BLOQUEIO DE REMETENTE DE DOMÍNIO DESCONHECIDO.”	107
FIGURA 52: “BLOQUEIO DE REMETENTE DE DOMÍNIO DE DNS REVERSO CONFIGURADO.”	108
FIGURA 53: “MENSAGEM DESCARTADA DEVIDO A DESTINATÁRIO INEXISTENTE NA BASE DE DADOS.”	108
FIGURA 54: “MENSAGEM BLOQUEADA PELO FILTRO SPF.”	109
FIGURA 55: “MENSAGEM BLOQUEADA PELO FILTRO POSTGREY.”	110
FIGURA 56: “FILTRO DO SOFTWARE DSPAM SEM TREINAMENTO.”	111
FIGURA 57: “FILTRO DO SOFTWARE DSPAM JÁ TREINADO.”	111

LISTAS DE QUADROS E TABELAS

QUADRO 1: “PRINCIPAIS COMANDOS SMTP.”	21
QUADRO 2: “PRINCIPAIS COMANDOS POP”	24
QUADRO 3: “EXEMPLOS DE COMANDOS IMAP”	28
QUADRO 4: “ESQUEMA DE PARTIÇÕES CRIADO PARA O PROJETO.”	44
QUADRO 5: “PARÂMETROS DE AUTENTICAÇÃO PARA INTERFACE WEB DO DSPAM.”	76
QUADRO 6: “FUNÇÕES PRESENTES NO ARQUIVO FUNCTIONS.”	91
QUADRO 7: “VARIÁVEIS DO SCRIPT BENVINDO.SH.”	94
QUADRO 8: “FUNÇÕES DO SCRIPT BENVINDO.SH.”	96
TABELA 1: “TEMPOS OCORRIDOS DURANTE OS TESTES DE INSTALAÇÃO.”	112

LISTA DE ABREVIATURAS E SIGLAS

BASH - *Bourne Again Shell*

DNS - *Domain Name System*

ESMTP - *Extended Simple Mail Transfer Protocol*

FQDN - *Fully Qualified Domain Name*

GNU - *GNU is Not Unix*

IMAP - *Internet Message Access Protocol*

IP - *Internet Protocol*

KSH - *Korn Shell*

MDA - *Mail Delivery Agent*

MIME - *Multipart Internet Mail Extensions*

MRA - *Mail Retrieval Agent*

MTA - *Mail Transport Agent*

MUA - *Mail User Agent*

PHP - *Hypertext Preprocessor*

POP - *Post Office Protocol*

RFC - *Request For Comments*

RPM - *RPM Package Manager*

SASL - *Simple Authentication and Security Layer*

SGBD - *Sistema de Gerenciamento de Banco de Dados*

SMTP - *Simple Mail Transfer Protocol*

SPF - *Sender Policy Framework*

SQL - *Structured Query Language*

SSL - *Secure Sockets Layer*

TCP - *Transmission Control Protocol*

SUMÁRIO

1. INTRODUÇÃO	11
1.1. <i>Objetivo Geral</i>	11
1.2. <i>Objetivos específicos</i>	11
1.3. <i>Escopo do projeto</i>	12
1.4. <i>Organização do trabalho</i>	13
2. APRESENTAÇÃO DOS CENÁRIOS	14
2.1. <i>Problema apresentado e cenário atual</i>	14
2.2. <i>Proposta de solução</i>	15
3. TECNOLOGIAS UTILIZADAS NA SOLUÇÃO	16
3.1. <i>Conceitos Básicos</i>	16
3.1.1. Virtualização	16
3.1.2. O Banco de Dados PostgreSQL	17
3.2. <i>Sobre o serviço de e-mail</i>	18
3.2.1. Funcionamento do sistema de e-mail	18
3.2.2. Protocolo SMTP	20
3.2.3. Protocolos POP e IMAP	22
3.2.4. Tipos de caixa postal	30
3.2.4.1. MBOX	30
3.2.4.2. MAILDIR	31
3.3. <i>Conceitos de criptografia</i>	32
3.3.1. Algoritmos de chaves privadas e públicas	33
3.4. <i>DNS e o sua utilização nos sistemas de e-mail</i>	34
4. DESENVOLVIMENTO DO PROJETO	36
4.1. <i>Sobre os softwares escolhidos</i>	36
4.2. <i>Implementação manual do servidor</i>	39
4.2.1. Cenário proposto	39
4.2.2. Instalação do Mandriva Linux 2009.0	42
4.2.3. Implementação do servidor de e-mail	47
4.2.3.1. Configuração do banco de dados e do Postfix Admin	47
4.2.3.1. Configuração dos serviços de e-mail e do webmail	53
4.2.4. Implementação dos recursos de segurança	63
4.2.4.1. Restrições de recebimento de mensagens no Postfix e o SASL	63
4.2.4.2. Configuração do SPF e do Postgrey	65
4.2.4.3. Configuração do DSPAM	70
4.2.4.4. Utilização dos protocolos HTTPS, POPS e IMAPS	78
4.3. <i>Remasterização da imagem do sistema</i>	81
4.4. <i>Automatização da configuração do servidor</i>	88
5. RESULTADOS OBTIDOS	98
5.1. <i>Funcionamento do servidor</i>	98
5.2. <i>Testes dos recursos de segurança</i>	103
5.3. <i>Resultados da remasterização</i>	112
6. CONCLUSÃO	115

1. INTRODUÇÃO

A idéia do projeto inicia-se a partir da necessidade de tornar menos complexa uma das tarefas que exigem mais de um administrador na área de servidores GNU/Linux: a configuração de servidores de correio eletrônico.

1.1. Objetivo Geral

O principal objetivo desse projeto visa apresentar uma solução para o serviço de e-mail, que possa ser instalada, configurada e administrada de forma simples e prática, além de possuir recursos de segurança. Essa solução possui como base o sistema operacional GNU/Linux, com a distribuição Mandriva Linux 2009.

Para isso, foi criado um sistema modificado, em que partes desnecessárias do mesmo foram retiradas, incluindo somente softwares necessários ao funcionamento da solução. Além disso, foram criados scripts que automatizam grande parte do processo de instalação e configuração do sistema.

A segurança da solução também possui destaque. Foram integradas diversas ferramentas com a finalidade de reduzir os problemas de segurança relacionados ao serviço de e-mail, como por exemplo, os spams.

O produto final gerado pelo projeto é uma imagem de instalação remasterizada, voltada para o serviço de correio eletrônico, que tem como principal característica a facilidade de instalação e configuração, além de possuir recursos de segurança e ferramentas que facilitem a administração dos serviços utilizados.

1.2. Objetivos específicos

Os objetivos específicos do projeto são:

- Implementar um servidor de correio eletrônico utilizando um banco de dados para o armazenamento das contas de e-mail;
- Configurar filtros que auxiliam o combate às mensagens indesejadas, também conhecidas como spams;
- Instalar softwares de antivírus e antispam para verificação do corpo da mensagem recebida, além dos anexos existentes na mesma;
- Manter o sigilo das informações;
- Modificar a imagem de instalação do sistema escolhido com o intuito de automatizar o processo de instalação do mesmo;
- Criar scripts com o objetivo de configurar todo o servidor de forma simples e prática.

1.3. Escopo do projeto

O desenvolvimento do projeto foi dividido em três partes. A primeira etapa foi o processo de pesquisa e implementação da solução de forma manual. Nessa etapa foram instaladas e configuradas todas as ferramentas necessárias ao funcionamento completo da solução.

Já a segunda etapa trata do processo de remasterização da imagem de instalação do sistema. Nessa etapa foram retirados todos os softwares desnecessários ao funcionamento do servidor. Além disso, foi criado um script que automatiza todo o processo de instalação do sistema. Esse script foi incorporado na imagem de instalação. Por fim, foi adicionada uma estrutura de diretórios na imagem de instalação, contendo todos os arquivos necessários para a configuração do servidor. Esses arquivos serão utilizados pelo script que irá automatizar o processo de configuração do servidor.

Por fim, a terceira etapa trata da criação do script que automatiza todo o processo de configuração do servidor. Esse script utiliza os arquivos criados na segunda etapa para configurar de forma rápida e prática grande parte dos serviços do servidor.

As configurações do serviço DNS, necessárias para o funcionamento do servidor não fazem parte do escopo do projeto. Essas configurações foram realizadas para possibilitar a implementação da solução e os testes subsequentes, porém, não serão abordadas na monografia.

1.4. Organização do trabalho

No capítulo 2 é apresentado o problema a ser resolvido neste projeto. Além disso, é apresentada a proposta de solução para o problema apresentado.

Já no capítulo 3 são descritas as tecnologias utilizadas na implementação. Os tópicos desse capítulo servem como base para o entendimento do projeto.

No capítulo 4 é descrito todo o processo realizado durante a implementação da solução proposta para o projeto.

No capítulo 5 são mostrados os testes realizados com a solução implementada. Dentre eles estão incluídos os testes para verificar o funcionamento do servidor e dos recursos de segurança. Ao final do capítulo é mostrada uma análise feita com a finalidade de mostrar as vantagens trazidas pela automatização da instalação e configuração do servidor.

Por fim, no capítulo 6 é apresentada a conclusão do projeto. Além disso, são propostas idéias para trabalhos futuros.

2. APRESENTAÇÃO DOS CENÁRIOS

2.1. Problema apresentado e cenário atual

A configuração dos servidores de correio eletrônico é um verdadeiro desafio, pois são necessárias diversas ferramentas - tanto para fornecer recursos que um servidor de e-mail robusto necessita, como também para prover a segurança adequada para que os dados que ali trafegam estejam protegidos contra pessoas ou programas maliciosos - que para serem integradas de forma correta, precisam de administradores com bom nível técnico no assunto. De acordo com Dent (2003, p. 19), o serviço de e-mail para a internet é construído através de diversos padrões e protocolos que definem como as mensagens são criadas e transferidas de um remetente para um destinatário, sendo necessária a utilização de diversas ferramentas, cada uma com o seu papel durante o processo de entrega da mensagem.

Mesmo dispondo de uma pessoa capacitada para realizar esse tipo de configuração, essa provavelmente irá demandar um longo tempo para finalizar a implementação desse servidor, visto que além dos serviços de e-mail propriamente dito - o envio e o recebimento de e-mail de forma correta – ela terá que agregar diversas ferramentas que acrescentem recursos extras ao servidor, tanto para facilitar a administração do mesmo como também torná-lo seguro o suficiente que os dados manipulados estejam protegidos.

Diante dos grandes desafios que cercam a integração dos serviços que compõem um servidor de e-mail, alguns administradores optam por excluir vários recursos do mesmo, tornando a implementação funcional. Porém essa decisão pode acarretar na difícil administração do servidor, ou também no comprometimento dos dados, por exemplo.

A falta de uma distribuição GNU/Linux, preparada essencialmente com o propósito prover ferramentas que automatizem todo o processo da instalação e

configuração faz com que a implementação de serviços de correio eletrônico seja demorada e difícil.

2.2. Proposta de solução

A partir da análise dos problemas encontrados, a proposta desse projeto é apresentar uma solução completa para os serviços de e-mail, livre de custos com softwares proprietários, que possa ser instalada, configurada e administrada de forma simples e prática, além de possuir recursos de segurança. Essa solução possui como base o sistema operacional GNU/Linux, com a distribuição Mandriva Linux 2009.

A distribuição Mandriva Linux é customizada para atender aos requisitos do servidor de e-mail. São inseridos na imagem remasterizada somente os pacotes relacionados ao funcionamento do servidor de e-mail. Além disso, todo o processo de instalação do servidor ocorre de forma automatizada e rápida.

O servidor criado no projeto tem como característica o suporte a domínios virtuais, ou seja, no mesmo servidor podem coexistir vários domínios. O servidor também possui uma interface web para a administração dos domínios e contas de e-mail.

A implementação realizada no projeto busca garantir a segurança do servidor, incorporando diversas ferramentas atualizadas que juntas buscam assegurar que os dados manipulados pelo servidor estão seguros.

3. TECNOLOGIAS UTILIZADAS NA SOLUÇÃO

Neste capítulo são descritas as tecnologias utilizadas durante a implementação do projeto.

3.1. Conceitos Básicos

3.1.1. Virtualização

A virtualização é uma tecnologia que permite a execução de diversos sistemas operacionais, independentes entre si, em um único equipamento. De acordo com a *Hewlett-Packard*, uma máquina virtual é um ambiente operacional completo que se comporta como se fosse um computador independente. [Hewlett-Packard, 2009]

Um dos grandes benefícios de se utilizar a virtualização é que se pode economizar gastos tanto de hardware como de energia substituindo servidores físicos por máquinas virtuais, todas rodando sob um mesmo sistema operacional em um único hardware. O produto final gerado é um conjunto de sistemas operacionais rodando simultaneamente em uma mesma máquina real. [MICROSOFT, 2009]

A figura 1 mostra de forma simples o processo de virtualização utilizado no projeto:



Figura 1: “Modelo de virtualização utilizado no projeto”.

Fonte: <http://www.microsoft.com/brasil/servidores/virtualizacao/solution-tech-server.msp>

3.1.2. O Banco de Dados PostgreSQL

O PostgreSQL é um SGBD objeto-relacional de código aberto, que teve o início de seu desenvolvimento em 1977. Ele começou em um projeto chamado *Ingres*, na universidade de *Berkeley*, na Califórnia - EUA. [WORSLEY; DRAKE, 2001]

Já em 1986, outro time de desenvolvedores da universidade de Berkeley continuou o desenvolvimento do código do Ingres para criar um SGBD objeto-relacional, que foi chamado de Postgres. Em 1996, devido ao grande avanço em suas funcionalidades, seu nome foi alterado para PostgreSQL. [WORSLEY; DRAKE, 2001]

Segundo Worsley e Drake (2001, p. 9), o PostgreSQL é mundialmente considerado o SGBD de código aberto mais avançado do mundo, que possui diversas características tradicionalmente encontradas somente em sistemas de banco de dados comerciais.

Algumas de suas principais características são:

1. Faz a abordagem dos dados utilizando um modelo objeto-relacional, com capacidade de manipular rotinas e regras complexas;
2. Suporte a sub-consultas;
3. Suporte a Integridade Referencial, que é utilizada para garantir a validade dos dados do banco de dados;
4. Suporte a Funções Armazenadas – que são geralmente denominadas Storage Procedures -, podendo ser escritas em diversas linguagens, como por exemplo o Perl e o Python;
5. Suporte a conexões SSL.

3.2. Sobre o serviço de e-mail

3.2.1. Funcionamento do sistema de e-mail

Para haver um fácil entendimento de como funciona um sistema de e-mail, alguns elementos, também chamados de agentes, devem ser discutidos: [WOOD, 1999]

- MUA (Mail User Agent): é o cliente de e-mail, utilizado pelo usuário para enviar e receber as mensagens. Um MUA pode ser também um programa ou um script que simula o comportamento do mesmo, enviando ou recebendo e-mails.
- MTA (Mail Transfer Agent): é o servidor de e-mail propriamente dito. É através dele que as mensagens são enviadas para a internet. A mensagem é transferida pela internet de um MTA de origem para um MTA de destino, através do protocolo SMTP, que será descrito mais adiante.
- MDA (Message Delivery Agent): é o programa – requisitado pelo MTA - responsável por escrever a mensagem na caixa postal do usuário, normalmente no sistema de arquivos do servidor.
- MRA (Mail Retrieval Agent): possui a responsabilidade de recuperar a mensagem, originalmente na caixa postal do usuário em um servidor remoto, para o MUA, que irá encarregar de exibir a mensagem. Esse processo de recuperação é realizado através dos protocolos POP e IMAP, que serão explicados mais adiante.

Além desses conceitos informados acima, é importante ressaltar que o sistema de e-mail utiliza o MIME (Multipart Internet Mail Extensions) para anexar arquivos às mensagens. O MIME é uma série de especificações que descrevem como se devem representar dados binários em texto puro para que esses possam ser enviados via correio eletrônico. [WOOD, 1999]

Basicamente, o funcionamento de um sistema de e-mail, desde o envio até recebimento, pode ser descrito da seguinte forma:

1. O cliente de e-mail (MUA) envia a mensagem para o servidor (MTA) que está configurado para o envio de e-mails. Esse é conhecido como o MTA remetente. Protocolo utilizado nessa fase: SMTP;
2. O MTA remetente envia a mensagens para um ou mais MTAs de destino, chamados de MTAs destinatários. Esses possuem uma base de dados – armazenada em um banco de dados, por exemplo – das contas de e-mail dos usuários. Protocolo utilizado nessa fase: SMTP;
3. O MTA destinatário transfere a mensagem para o MDA, que irá gravar a mensagem na caixa postal do usuário. Essa mensagem geralmente é um arquivo no sistema de arquivos do servidor. Todo o processo é local, ou seja, ocorre no MTA destinatário;
4. O usuário destinatário utiliza um cliente de e-mail para baixar a mensagem, através do MRA. Protocolo utilizado nessa fase: POP ou IMAP.

A figura 2 mostra o funcionamento do envio de uma mensagem:

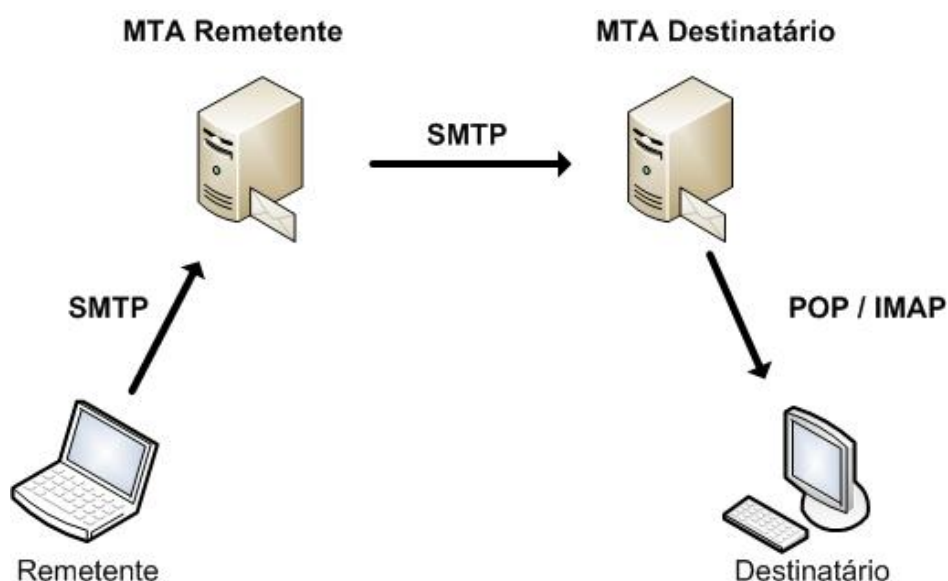


Figura 2: "Funcionamento do sistema de e-mail".

Uma observação importante: um MTA pode ser tanto remetente como destinatário, ao enviar ou receber e-mails, respectivamente. Os protocolos SMTP, POP e IMAP, citados acima, serão descritos nos próximos tópicos.

3.2.2. Protocolo SMTP

Segundo Wood (1999, p. 117) o SMTP (*Simple Mail Transfer Protocol*) e suas extensões, criadas durante o tempo de amadurecimento do protocolo, formam a base do sistema de e-mail. Ele é usado tanto pelos MUAs, como também pelos próprios MTAs, nas transferências de mensagens de correio eletrônico, conforme foi explicado no tópico anterior.

Um exemplo de MTA que fornece o serviço de SMTP é o Postfix, que será utilizado na implementação deste projeto. As extensões do protocolo SMTP - criadas durante o amadurecimento do mesmo, conforme já dito - são chamadas de ESMTP (*Extended Simple Mail Transfer Protocol*) e são definidas pela RFC¹ 1869. [WOOD, 1999]

O SMTP é utilizado entre um cliente, que deseja enviar uma mensagem; e um MTA, que irá aceitá-la para entrega. O servidor poderá entregar a mensagem para si próprio ou para outro MTA, dependendo dos seguintes fatores:

- Entrega para si mesmo, caso o destinatário faça parte de um domínio que está sob controle do servidor. Exemplo: caso o domínio sob controle do MTA seja **exemplo.com.br** e o destinatário seja **user@exemplo.com.br**;
- Envia para o MTA responsável, caso o destinatário não faça parte de um domínio que está sob controle do servidor. Exemplo: caso o domínio sob controle do MTA seja **exemplo.com.br** e o destinatário seja **user@teste.com.br**.

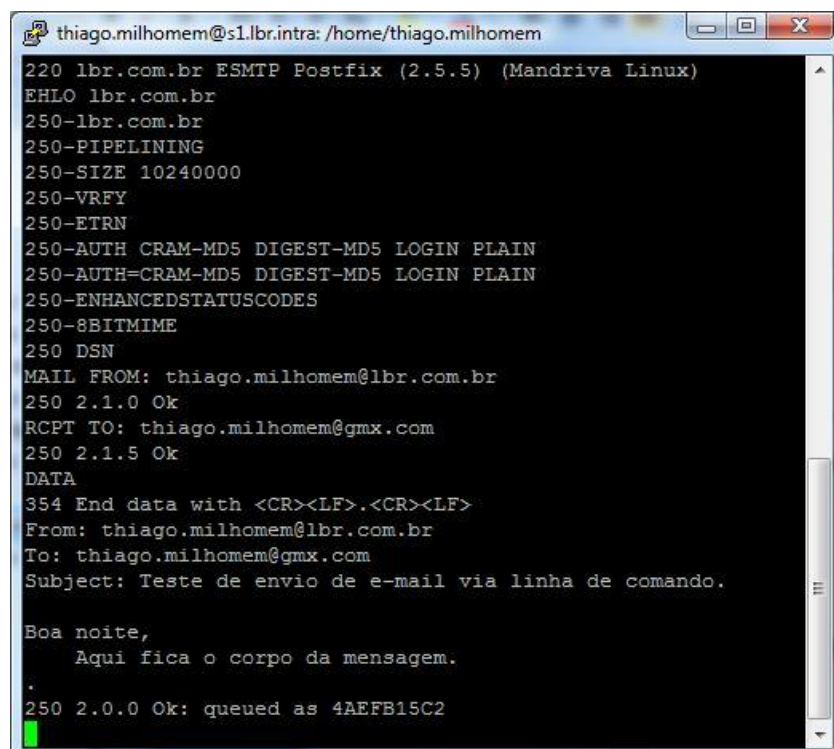
¹ Request for Comments. Disponível em: <<http://www.ietf.org/rfc.html>>. Acesso em 19 mar 2009.

A conversão entre agentes – sejam eles MUAs ou MTAs – é realizada através de comandos. Alguns dos principais comandos são descritos no quadro 1:

Quadro 1: “Principais comandos SMTP.”

Comando	Argumento passado	Descrição
EHLO	Nome do domínio	Utilizado para identificar o cliente e listar as extensões ESMTP suportadas pelo servidor.
MAIL	E-mail	Informa o servidor da conta de e-mail do remetente.
RCPT	E-mail	Informa o servidor da conta de e-mail do destinatário.
DATA	Conteúdo da mensagem	Escreve a mensagem e envia para o servidor.

A figura 3 demonstra o envio de uma mensagem através de um terminal de comando:



```

thiago.milhomem@sl.lbr.intra: /home/thiago.milhomem
220 lbr.com.br ESMTP Postfix (2.5.5) (Mandriva Linux)
EHLO lbr.com.br
250-lbr.com.br
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-AUTH CRAM-MD5 DIGEST-MD5 LOGIN PLAIN
250-AUTH=CRAM-MD5 DIGEST-MD5 LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: thiago.milhomem@lbr.com.br
250 2.1.0 Ok
RCPT TO: thiago.milhomem@gmx.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: thiago.milhomem@lbr.com.br
To: thiago.milhomem@gmx.com
Subject: Teste de envio de e-mail via linha de comando.

Boa noite,
    Aqui fica o corpo da mensagem.
.
250 2.0.0 Ok: queued as 4AEFB15C2
  
```

Figura 3: “Envio de e-mail via linha de comando”.

3.2.3. Protocolos POP e IMAP

Basicamente existem dois tipos de mecanismos para a busca de correio eletrônico em servidores remotos: o POP e o IMAP. O primeiro é o protocolo mais simples e antigo dos dois. O POP é um dos dois mecanismos que implementa o conceito dos MRAs, que foi descrito em tópicos anteriores. Ele é um sistema do tipo cliente/servidor e utiliza a porta TCP 110 como padrão. Sua versão atual é chamada POP3. [WOOD, 1999]

O protocolo POP3 atua no estágio em que o cliente de e-mail requisita a busca de uma mensagem no servidor. Ele provê mecanismos com a finalidade de realizar essa tarefa. Recapitulando de forma breve os estágios em que o e-mail é entregue ao MTA destinatário, o processo ocorre da seguinte forma:

1. MTA destinatário recebe a mensagem do MTA remetente;
2. O MTA destinatário entrega a mensagem ao MDA, que por sua vez grava a mensagem na caixa postal do usuário;
3. O MRA, quando requisitado, utiliza o protocolo POP3 ou IMAP para entregar a mensagem para o cliente de e-mail (*MUA*) do usuário.

Um cliente que possui a capacidade de fornecer o suporte ao protocolo POP3 é denominado como cliente POP3. A figura 4 demonstra o funcionamento do estágio descrito acima:

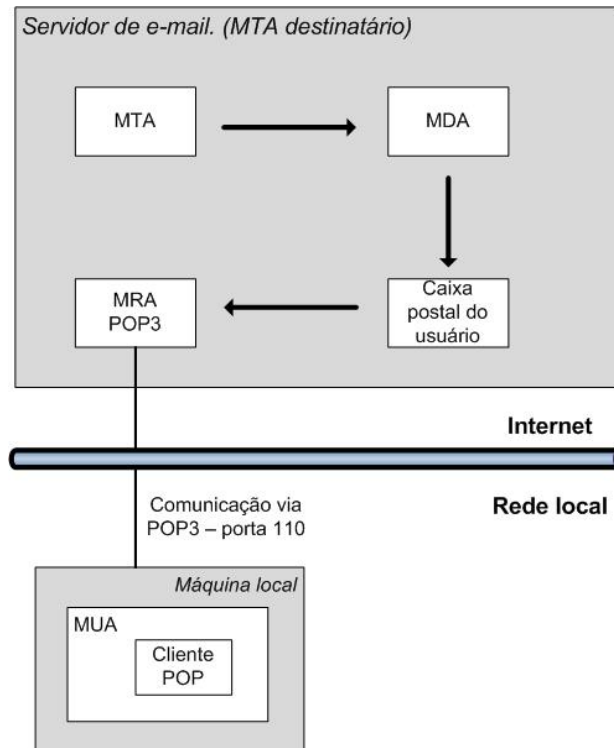


Figura 4: “Funcionamento da entrega de uma mensagem”.

É importante frisar que todas as conexões POP são originadas pelo cliente. O papel do servidor POP3 (MRA) é aguardar por requisições e atendê-las quando solicitadas. Vale ressaltar também que, as mensagens não são manipuladas no servidor, ou seja, um cliente busca as mensagens – escolhendo se deseja manter uma cópia ou apagá-la do servidor – para depois fazer as manipulações desejadas.

Uma sessão POP3 – processo de comunicação entre cliente e servidor via POP3 – possui três estados:

1. Estado de autorização;
2. Estado de transação;
3. Estado de atualização.

A comunicação inicia-se com o cliente estabelecendo a conexão na porta TCP 110. Feita a conexão, o servidor envia uma mensagem de boas-vindas e a sessão está iniciada. A partir desse ponto, a comunicação acontece da seguinte maneira:

1. O cliente POP3 especifica qual será o mecanismo de autenticação que irá utilizar. Alguns exemplos de mecanismos são: PLAIN, LOGIN e DIGEST-MD5. Este estágio caracteriza o **estado de autorização**;
2. O servidor, caso suporte o mecanismo escolhido, solicita o usuário e senha; porém, se não possuir suporte, envia uma mensagem de erro para o cliente. Estágio ainda no **estado de autorização**;
3. Após o acerto do mecanismo de autenticação com o servidor, o cliente fornece o usuário e senha de sua conta. Logo após isso, o servidor verifica as credenciais e se as mesmas estiverem corretas, trava a caixa postal para o cliente conectado – evitando assim, que outros clientes tenham acesso naquele momento - iniciando assim, o **estado de transação**;
4. Durante o **estado de transação**, o cliente executa comandos específicos para manipular as mensagens em sua caixa postal.
5. O estado de transação termina quando o cliente executa o comando *QUIT*. A partir daí, inicia-se o **estado de atualização**, no qual o servidor deleta as mensagens que foram marcadas para a remoção. Além disso, o servidor também retira a trava da caixa postal que foi criada no início do **estado de transação**.

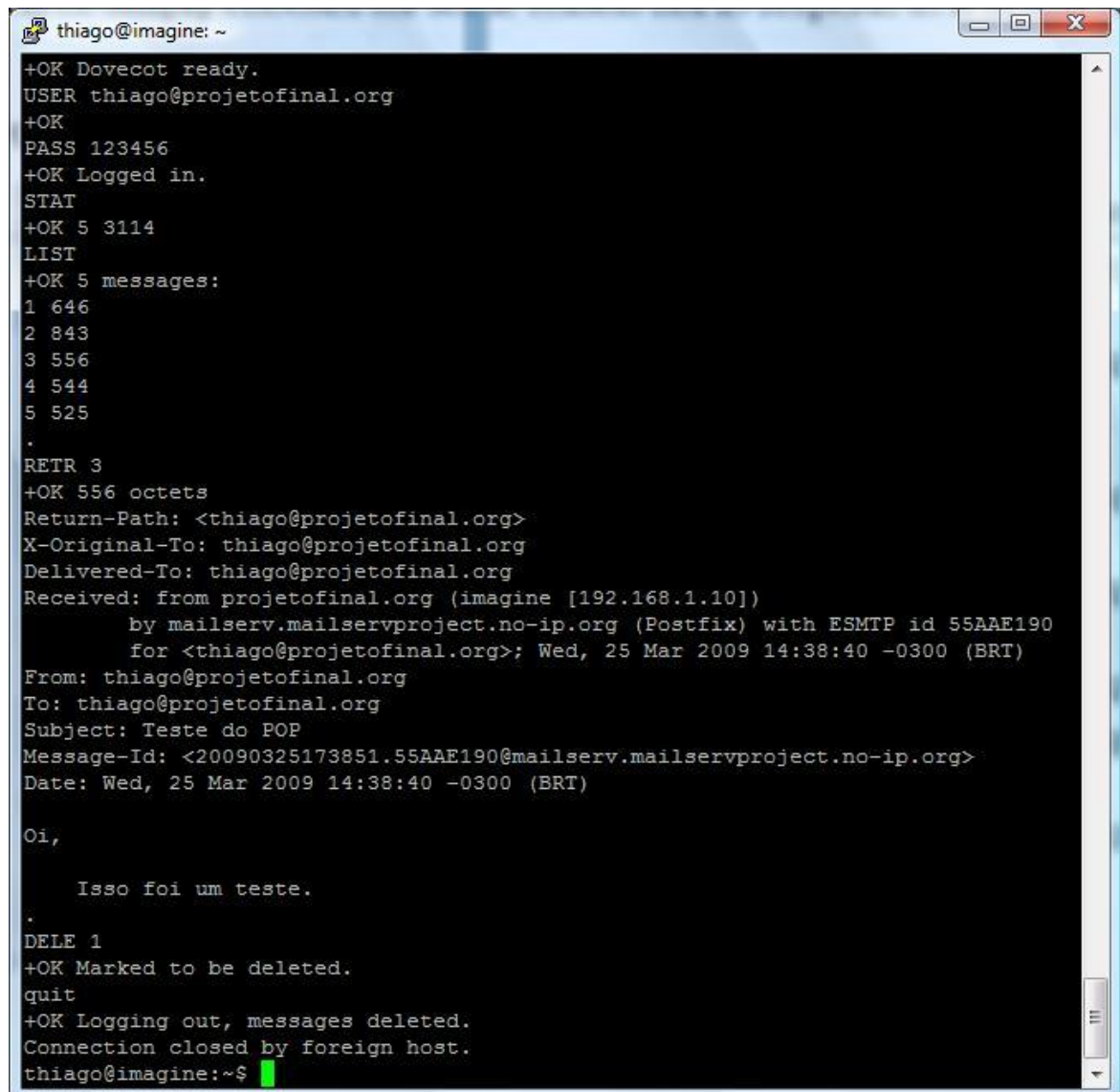
A partir do quadro 2 é possível observar os principais comandos durante uma comunicação POP:

Quadro 2: “Principais comandos POP”

Comando	Argumento passado	Descrição
Estado de autorização		
AUTH	Mecanismo de autenticação	Especifica ao servidor qual mecanismo de autenticação irá utilizar.
USER	Usuário da caixa postal	Especifica qual é o usuário que irá se autenticar.

PASS	Senha do usuário	Informa ao servidor qual a senha do usuário.
Estado de transação		
STAT		Solicita ao servidor o estado atual da caixa postal. Informa ao cliente quantas mensagens existem.
LIST		Realiza uma listagem das mensagens da caixa postal.
RETR	Número da mensagem	Exibe a mensagem solicitada.
DELE	Número da mensagem	Marca uma mensagem para exclusão.

A figura 5 mostra, via linha de comando, um exemplo de sessão POP3:



```

thiago@imagine: ~
+OK Dovecot ready.
USER thiago@projetofinal.org
+OK
PASS 123456
+OK Logged in.
STAT
+OK 5 3114
LIST
+OK 5 messages:
1 646
2 843
3 556
4 544
5 525
.
RETR 3
+OK 556 octets
Return-Path: <thiago@projetofinal.org>
X-Original-To: thiago@projetofinal.org
Delivered-To: thiago@projetofinal.org
Received: from projetofinal.org (imagine [192.168.1.10])
        by mailserv.mailservproject.no-ip.org (Postfix) with ESMTP id 55AAE190
        for <thiago@projetofinal.org>; Wed, 25 Mar 2009 14:38:40 -0300 (BRT)
From: thiago@projetofinal.org
To: thiago@projetofinal.org
Subject: Teste do POP
Message-Id: <20090325173851.55AAE190@mailserv.mailservproject.no-ip.org>
Date: Wed, 25 Mar 2009 14:38:40 -0300 (BRT)

Oi,

    Isso foi um teste.
.
DELE 1
+OK Marked to be deleted.
quit
+OK Logging out, messages deleted.
Connection closed by foreign host.
thiago@imagine:~$

```

Figura 5: "Sessão POP3 via linha de comando".

Já o protocolo IMAP (*Internet Message Access Protocol*), criado para ser o sucessor do POP, possui a capacidade de administrar a caixa postal remotamente, mantendo as mensagens no servidor. De acordo com Wood (1999, p. 151), as caixas postais IMAP podem ser acessadas por diversos clientes, visto que as mensagens são mantidas no servidor. A figura 6 exemplifica esse conceito:

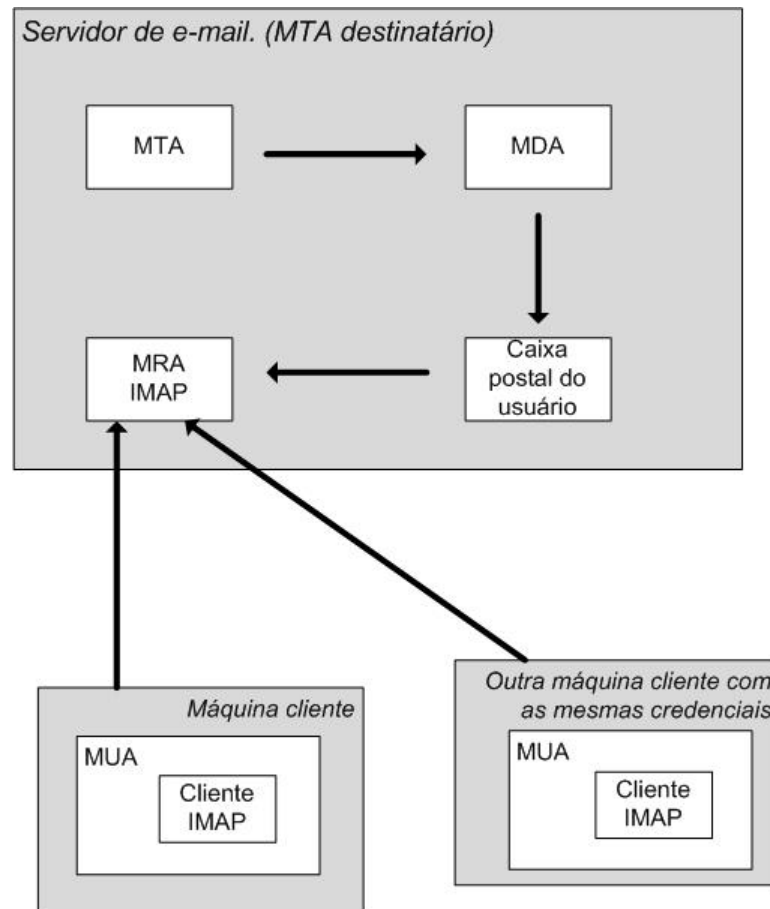


Figura 6: "Funcionamento do IMAP".

Um servidor IMAP – considerado um MRA, assim como o servidor POP – escuta por solicitações na porta TCP 143, conhecida como porta padrão para o IMAP. Clientes que desejam conectar-se no servidor IMAP deverão utilizar a porta citada acima. Um cliente que possui a capacidade de se comunicar via protocolo IMAP é denominado um cliente IMAP. A grande maioria dos clientes que suportam o protocolo IMAP também possuem suporte ao POP3. [WOOD, 1999]

Um cliente IMAP pode tanto buscar as mensagens para depois manipulá-las, emulando assim o POP, como também pode realizar as manipulações no próprio servidor, sem a necessidade de carregá-las para o computador. [WOOD, 1999] Esse último método citado é interessante para um usuário que deseja acessar sua conta de e-mail através de seu computador pessoal, seu notebook e seu celular, supondo que todos esses possuem clientes IMAP, por exemplo.

Assim como o POP, durante uma sessão IMAP, existem estados distintos. São eles:

1. **Estado não-autenticado:** estado onde o cliente ainda não foi autenticado no servidor;
2. **Estado autenticado:** estado onde o cliente fornece, de forma correta, suas credenciais ao servidor. Uma vez autenticado, o cliente pode manipular diretórios IMAP, como por exemplo:
 - a. Criar ou apagar diretórios;
 - b. Selecionar um diretório para manipular as mensagens contidas nele;
 - c. Realizar a inscrição ou desinscrição de diretórios. Bastante útil quando se possui diversos diretórios IMAP, porém somente deseja exibir alguns, por exemplo.
3. **Estado de seleção:** estado em que o cliente seleciona um diretório IMAP com a finalidade de manipular as mensagens contidas nesse diretório;
4. **Estado de *logout*:** ocorre quando o cliente termina a conexão, quando o servidor recusa a conexão ou quando a conexão é interrompida.

O servidor, durante uma sessão IMAP, aguarda comandos específicos para cada estado acima citado. Seguem alguns exemplos no quadro 3:

Quadro 3: “Exemplos de comandos IMAP”.

Comando	Parametros passados	Descrição
Estado não-autenticado		
LOGIN	Usuário e senha	Solicita autenticação ao servidor.
Estado autenticado		
SELECT	Diretório IMAP	Acessa um determinado diretório IMAP.
CREATE	Diretório IMAP	Cria um novo diretório IMAP com o nome especificado.
DELETE	Diretório IMAP	Apaga o diretório IMAP especificado.
RENAME	Diretório IMAP	Renomeia o diretório IMAP especificado.
SUBSCRIBE	Diretório IMAP	Adiciona o diretório IMAP especificado à lista de diretórios inscritos.
UNSUBSCRIBE	Diretório IMAP	Retira o diretório IMAP especificado da lista de diretórios inscritos.
LSUB	Diretório IMAP	Lista os diretórios IMAP que estão na lista inscritos.
Estado seleção		
COPY	<Número da mensagem> <diretório IMAP>	Copia uma mensagem para outro diretório IMAP.
FETCH	Número da mensagem	Exibe a mensagem especificada.

As figuras 7 e 8 mostram alguns comandos utilizados em uma sessão IMAP:

```

thiago@imagine: ~
* OK Dovecot ready.
. LOGIN thiago@projetofinal.org 123456
. OK Logged in.
. LSUB "" ""
* LSUB () "." "INBOX"
* LSUB () "." "INBOX.Pessoal"
* LSUB () "." "INBOX.Faculdade"
* LSUB () "." "INBOX.Trash"
. OK Lsub completed.
. CREATE INBOX.Diversos
. OK Create completed.
. SUBSCRIBE INBOX.Diversos
. OK Subscribe completed.
. LSUB "" ""
* LSUB () "." "INBOX"
* LSUB () "." "INBOX.Pessoal"
* LSUB () "." "INBOX.Faculdade"
* LSUB () "." "INBOX.Trash"
* LSUB () "." "INBOX.Diversos"
. OK Lsub completed.
. UNSUBSCRIBE INBOX.Diversos
. OK Unsubscribe completed.
. LSUB "" ""
* LSUB () "." "INBOX"
* LSUB () "." "INBOX.Pessoal"
* LSUB () "." "INBOX.Faculdade"
* LSUB () "." "INBOX.Trash"
. OK Lsub completed.

```

Figura 7: "Comandos IMAP no estado autenticado".

```

thiago@imagine: ~
* OK Dovecot ready.
. LOGIN thiago@projetofinal.org 123456
. OK Logged in.
. SELECT INBOX.Faculdade
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags permitted.
* 2 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1236038409] UIDs valid
* OK [UIDNEXT 3] Predicted next UID
. OK [READ-WRITE] Select completed.
. FETCH 1 RFC822
* 1 FETCH (RFC822 {714}
Return-Path: <thiago@projetofinal.org>
X-Original-To: thiago@projetofinal.org
Delivered-To: thiago@projetofinal.org
Received: from 192.168.1.4 (mailserv [127.0.0.1])
    by mailserv.mailservproject.no-ip.org (Postfix) with
    ESMTP id BAlAD125
    for <thiago@projetofinal.org>; Wed, 25 Mar 2009 17:2
0:04 -0300 (BRT)
MIME-Version: 1.0
Date: Wed, 25 Mar 2009 17:20:04 -0300
From: <thiago@projetofinal.org>
To: Eu <thiago@projetofinal.org>
Subject: Estudo faculdade 1
Message-ID: <e48dd826b8a878ec70ca8d5b1e604358@localhost>
X-Sender: thiago@projetofinal.org
User-Agent: RoundCube Webmail/0.2
Content-Transfer-Encoding: 8bit
Content-Type: text/plain; charset="UTF-8"

Oi,

    Esse foi o primeiro estudo.

)
. OK Fetch completed.
. COPY 1 INBOX.Pessoal
. OK [COPYUID 1236038410 1 2] Copy completed.

```

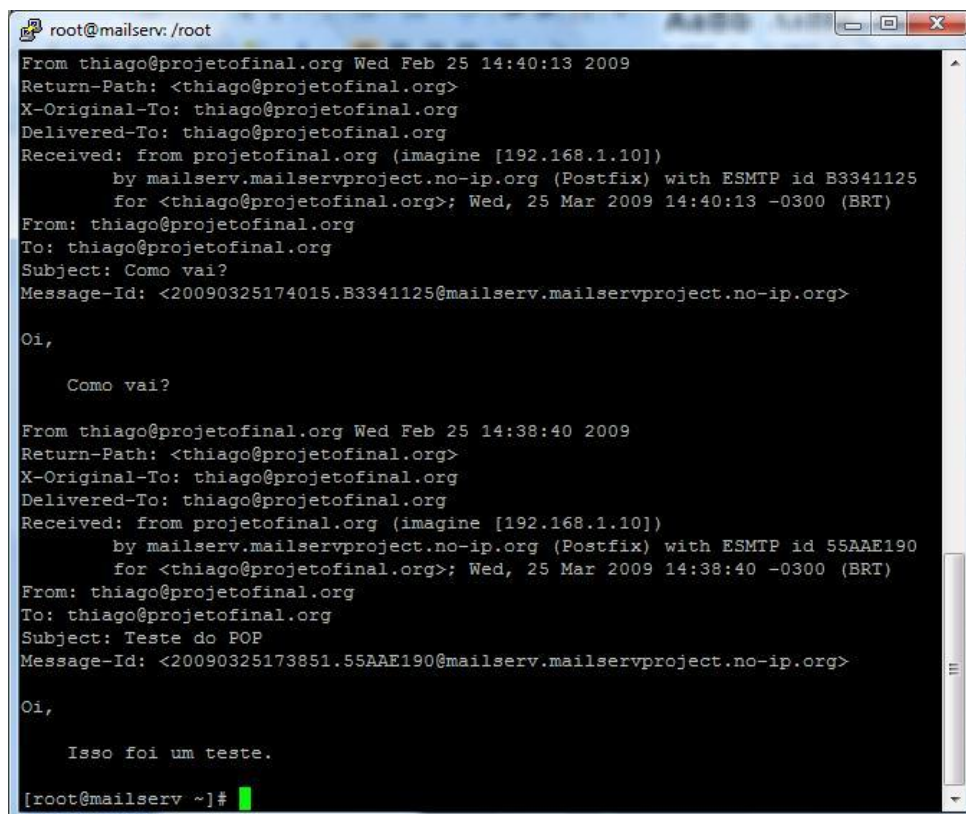
Figura 8: "Comandos IMAP no estado de seleção".

3.2.4. Tipos de caixa postal

Os tipos de caixa postal estão relacionados à forma em que as mensagens são escritas pelo MDA no sistema de arquivos do servidor. Os dois tipos mais conhecidos são o MBOX e o MAILDIR.

3.2.4.1. MBOX

O formato MBOX foi desenvolvido sob a plataforma UNIX e também é chamado de “formato UNIX”. Seu método de armazenamento de mensagens consiste em concatenar todas as mensagens em um único arquivo. O separador das mensagens concatenadas é chamado de linha “*From_*”. Essa linha é constituída da palavra *From*, seguida de um espaço e o endereço de e-mail do remetente, além da data da mensagem. A figura 9 mostra uma caixa postal no formato MBOX:



```

root@mailserv: /root
From thiago@projetofinal.org Wed Feb 25 14:40:13 2009
Return-Path: <thiago@projetofinal.org>
X-Original-To: thiago@projetofinal.org
Delivered-To: thiago@projetofinal.org
Received: from projetofinal.org (image [192.168.1.10])
        by mailserv.mailservproject.no-ip.org (Postfix) with ESMTP id B3341125
        for <thiago@projetofinal.org>; Wed, 25 Mar 2009 14:40:13 -0300 (BRT)
From: thiago@projetofinal.org
To: thiago@projetofinal.org
Subject: Como vai?
Message-Id: <20090325174015.B3341125@mailserv.mailservproject.no-ip.org>

Oi,

    Como vai?

From thiago@projetofinal.org Wed Feb 25 14:38:40 2009
Return-Path: <thiago@projetofinal.org>
X-Original-To: thiago@projetofinal.org
Delivered-To: thiago@projetofinal.org
Received: from projetofinal.org (image [192.168.1.10])
        by mailserv.mailservproject.no-ip.org (Postfix) with ESMTP id 55AAE190
        for <thiago@projetofinal.org>; Wed, 25 Mar 2009 14:38:40 -0300 (BRT)
From: thiago@projetofinal.org
To: thiago@projetofinal.org
Subject: Teste do POP
Message-Id: <20090325173851.55AAE190@mailserv.mailservproject.no-ip.org>

Oi,

    Isso foi um teste.

[root@mailserv ~]#

```

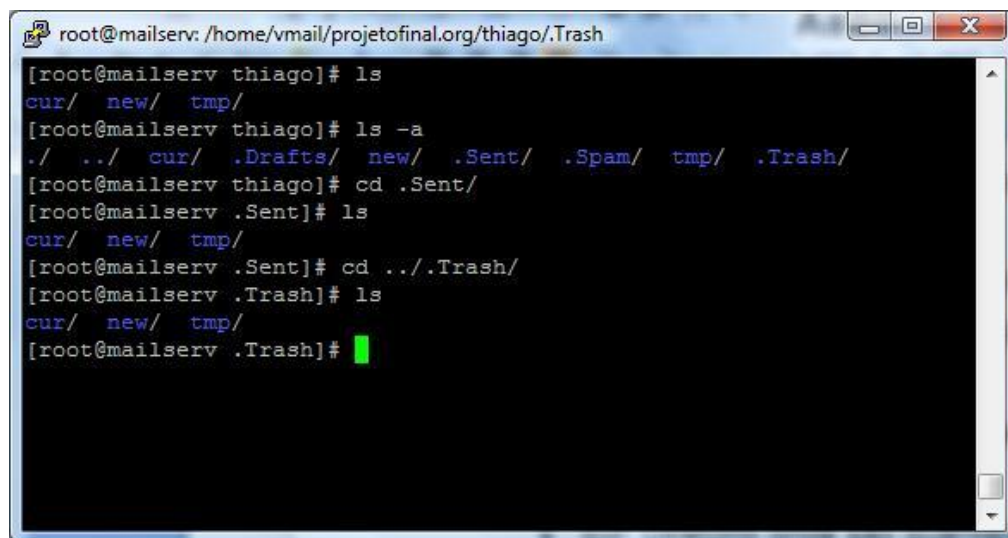
Figura 9: “Caixa postal no formato MBOX”.

3.2.4.2. MAILDIR

O formato MAILDIR foi criado inicialmente para ser utilizado com o MTA QMAIL². Diferente do MBOX, o MAILDIR utiliza diretórios para organizar as mensagens, sendo que cada uma dessas mensagens são guardadas em um arquivo único. Cada diretório possui três subdiretórios:

- tmp: Uma mensagem recém chegada é gravada neste diretório para depois ser movida para o diretório new.
- new: Diretório onde são guardadas as novas mensagens.
- cur: Diretório onde são guardadas as mensagens já lidas.

De acordo com a figura 10, é possível perceber que a caixa de entrada é o diretório pai e os outros diretórios – enviados, rascunhos, spam, lixeira – são iniciados por um ponto. Em sistemas *Unix-like*³ isso significa que esse diretório está oculto. Dentro de cada diretório podemos ver os subdiretórios *cur*, *new* e *tmp*.



```

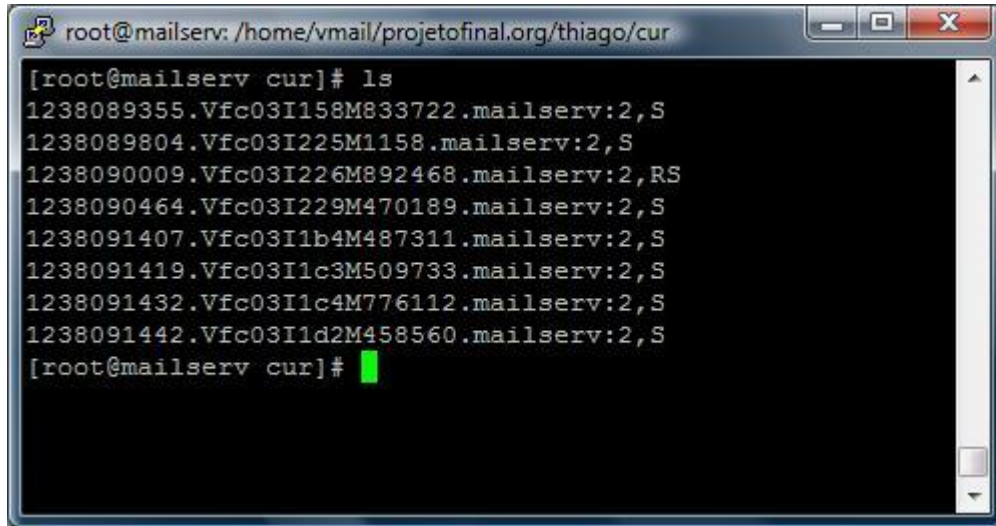
root@mailserv: /home/vmail/projetofinal.org/thiago/.Trash
[root@mailserv thiago]# ls
cur/  new/  tmp/
[root@mailserv thiago]# ls -a
./  ../  cur/  .Drafts/  new/  .Sent/  .Spam/  tmp/  .Trash/
[root@mailserv thiago]# cd .Sent/
[root@mailserv .Sent]# ls
cur/  new/  tmp/
[root@mailserv .Sent]# cd ../.Trash/
[root@mailserv .Trash]# ls
cur/  new/  tmp/
[root@mailserv .Trash]#
  
```

Figura 10: “Estrutura de diretórios do formato MAILDIR”.

2 MTA com o foco na estabilidade. Disponível em: <<http://www.qmail.org/top.html>>. Acesso em 26 mar 2009.

3 Sistemas operacionais que possuem como base o sistema Unix.

Já a figura 11 mostra como são organizadas as mensagens. No exemplo desta figura, as mensagens já foram lidas e, portanto, encontram-se no diretório *cur* da caixa de entrada:

A terminal window titled 'root@mailserv: /home/vmail/projetofinal.org/thiago/cur' displays the output of the 'ls' command. The output lists eight files in MAILDIR format, each on a new line. The files are: 1238089355.Vfc03I158M833722.mailserv:2,S; 1238089804.Vfc03I225M1158.mailserv:2,S; 1238090009.Vfc03I226M892468.mailserv:2,RS; 1238090464.Vfc03I229M470189.mailserv:2,S; 1238091407.Vfc03I1b4M487311.mailserv:2,S; 1238091419.Vfc03I1c3M509733.mailserv:2,S; 1238091432.Vfc03I1c4M776112.mailserv:2,S; and 1238091442.Vfc03I1d2M458560.mailserv:2,S. The prompt '[root@mailserv cur]#' is visible at the bottom of the terminal, followed by a green cursor.

```
root@mailserv: /home/vmail/projetofinal.org/thiago/cur
[root@mailserv cur]# ls
1238089355.Vfc03I158M833722.mailserv:2,S
1238089804.Vfc03I225M1158.mailserv:2,S
1238090009.Vfc03I226M892468.mailserv:2,RS
1238090464.Vfc03I229M470189.mailserv:2,S
1238091407.Vfc03I1b4M487311.mailserv:2,S
1238091419.Vfc03I1c3M509733.mailserv:2,S
1238091432.Vfc03I1c4M776112.mailserv:2,S
1238091442.Vfc03I1d2M458560.mailserv:2,S
[root@mailserv cur]#
```

Figura 11: “Organização das mensagens no formato MAILDIR”.

3.3. Conceitos de criptografia

A criptografia é considerada como fundamental para a segurança das organizações. O processo de cifragem disfarça a mensagem original, denominado texto claro (plaintext ou cleartext), de um modo que o resultado gera o texto cifrado (ciphertext). Já o processo de decifragem (decrypt) transforma de volta o texto cifrado para o texto claro. [NAKAMURA; GEUS, 2003] Esses processos, tanto de cifragem como de decifragem são realizados pelas chaves públicas e privadas. Esses conceitos serão discutidos logo adiante.

A criptografia garante as propriedades importantes para a proteção das informações. São elas:

1. Integridade;
2. Autenticidade;

3. Não-repúdio;

4. Sigilo

3.3.1. Algoritmos de chaves privadas e públicas

A criptografia de chave privada, – ou também, assimétrica – através da utilização de uma chave secreta para a codificação e decodificação dos dados, é responsável pelo sigilo das informações.

Uma das características dos algoritmos de chave simétrica é a rapidez na execução, porém esses algoritmos não permitem a assinatura e certificação digital. Além desse fato, existe o problema da transmissão dessa chave privada, visto que ainda não existe um canal seguro para o envio da chave. Outro problema encontrado é a necessidade do uso de chaves secretas diferentes para cada conexão, o que torna seu gerenciamento muito complexo. [NAKAMURA; GEUS, 2003]

Já os algoritmos de chave pública ou assimétrica podem garantir além do sigilo, a integridade, não-repúdio e autenticidade. Diferente das chaves simétricas, agora existe a possibilidade de utilizar a assinatura e a certificação digital. Neste caso, a comunicação é estabelecida através de dois pares de chaves diferentes, uma pública e outra privada. É importante ressaltar que com as chaves assimétricas, reduz-se o problema da troca de chaves, visto que não é necessário estabelecer um canal seguro para tal. Um exemplo seria uma mensagem ser cifrada utilizando a chave pública e decifrada através somente da chave privada correspondente, garantindo assim, o sigilo. Porém, os algoritmos de chave assimétrica são sensivelmente mais lentos que os algoritmos de chave simétrica. De acordo com Nakamura e Geus (2003, p. 290) os algoritmos de chave pública são cerca de 60 a 70 vezes mais lentos que os algoritmos simétricos.

Sendo assim, como ambos os algoritmos possuem vantagens e desvantagens, normalmente os dois são utilizados em conjunto. Um exemplo seria o SSL, que utiliza o algoritmo RSA para o estabelecimento de um canal seguro e o algoritmo RC4 para o sigilo da mensagem. [NAKAMURA; GEUS, 2003]

3.4. DNS e o sua utilização nos sistemas de e-mail

O DNS (Domain Name System) é um serviço cuja função é transformar nomes de hosts em endereços IP. Ele foi definido pela RFC 882 em 1983, introduzindo duas idéias chaves: os dados são distribuídos e nomenclatura dos nomes é hierárquica. Fazer com que os dados sejam distribuídos significa que cada sítio atualiza suas próprias informações e essas se tornam disponíveis quase imediatamente. Já a nomenclatura hierárquica previne que ocorram conflitos entre nomes de hosts. Exemplo: um sítio que possui o domínio **exemplo.com.br** pode possuir diversos nomes de hosts, desde que todos possuam como sufixo o domínio: **servidor1.exemplo.com.br**, **servidor2.exemplo.com.br**. [DENT, 2003]

Cada domínio possui no mínimo dois servidores de nomes. Esses possuem as informações sobre o domínio. Essas informações são chamadas de *resource records* ou registros DNS. Existem diversos tipos de registros DNS que indicam diferentes tipos de informações, como por exemplo: endereços IP, servidores de nomes e encaminhamento de e-mails. Os principais registros relacionados ao serviço de e-mail são: [DENT, 2003]

- Registro **A**: possui os mapeamentos dos nomes em endereços IP. É através desse tipo de registros que os nomes dos sítios são transformados em endereços IP;
- Registro **CNAME**: sua finalidade é criar apelidos, apontando para nome de hosts válidos. Exemplo: no endereço **www.exemplo.com.br**, o prefixo **www** pode ser um registro **CNAME** apontando para o nome de um servidor web: **servidor1.exemplo.com.br**.
- Registro **MX**: indica informações sobre o encaminhamento de e-mails. Especificam os endereços IP dos servidores que controlam as mensagens de e-mail domínio. É através desses registros que os servidores de e-mail encontram as informações necessárias para encaminhar as mensagens para outros domínios;

- Registro **PTR**: possui os mapeamentos reversos de endereços IP para nomes de hosts. Funciona de modo contrário aos registros do tipo **A**. Esse tipo de registro normalmente é utilizado pelos servidores de e-mail para verificar se o endereço IP que está tentando entregar uma mensagem possui um nome de host válido associado. Essa técnica é utilizada para reduzir a quantidade de spams recebidos.

É importante destacar dois aspectos relacionados ao serviço de e-mail e o DNS. Para o envio de mensagens, o servidor precisa acessar servidores DNS com a finalidade de realizar a resolução de nomes de hosts, além de buscar informações sobre o encaminhamento de e-mails. Já no recebimento das mensagens, os domínios correspondentes ao servidor de e-mail precisam estar configurados corretamente para que as mensagens sejam encaminhadas ao mesmo. [DENT, 2003]

4. DESENVOLVIMENTO DO PROJETO

4.1. Sobre os softwares escolhidos

Os critérios utilizados para a escolha dos softwares utilizados na implementação foram baseados nos seguintes fatores:

- Experiência de uso com o software;
- Facilidade na instalação;
- Facilidade na integração entre os softwares.

A distribuição Mandriva Linux foi escolhida por possuir ferramentas que facilitam a configuração de diversos recursos do sistema, como por exemplo, as configurações de redes. Diferente de outras distribuições, como o Debian⁴ GNU/Linux, o Mandriva Linux possui uma ferramenta, chamada **drakconnect**, que realiza a configuração de rede de acordo com o tipo de conexão a ser utilizada. Isso torna o processo de configuração do servidor mais rápida e prática, visto que menos tempo é gasto para configurações básicas. Além disso, o Mandriva possui diversos pacotes já compilados em seus repositórios de software, o que torna a instalação dos softwares simples, pois não é preciso realizar a compilação do código fonte dos mesmos. A versão 2009.0 foi escolhida por ser a mais atual no momento do início da implementação do projeto.

Um dos motivos para a escolha do software Postfix foi a sua facilidade de instalação e configuração. Para sua instalação, basta utilizar os pacotes já compilados disponíveis nos repositórios do software do Mandriva. Outro MTA bastante conhecido é o QMAIL, porém esse necessita que o seu código fonte seja compilado para o seu funcionamento. Outro motivo para a adoção do Postfix e a sua facilidade de integração com outras ferramentas, como por exemplo, o software anti-

⁴ Distribuição GNU/Linux que possui foco na estabilidade do sistema. Maiores informações em: <http://www.debian.org/>. Acesso em 26 mai 2009.

spam DSPAM. Foi escolhida a versão 2.5.5 por ser a mais atual para o Mandriva 2009.0.

O software Dovecot foi escolhido por possuir diversos recursos que vão além de um servidor POP3 e IMAP. O Dovecot pode ser facilmente integrado ao Postfix com a finalidade de prover autenticação para o envio de e-mails, assunto que será discutido no tópico 4.2.4.1. Ele pode ser integrado também ao software DSPAM, fazendo com mensagens indesejadas sejam direcionadas para locais específicos. Outro software com funcionalidade semelhante é o Courier, porém ele possui menos recursos quando comparado ao Dovecot. A versão 1.1.6 foi escolhida por ser a mais atual para a distribuição Mandriva utilizada.

Os filtros SPF e Postgrey foram escolhidos por serem de fácil integração com o Postfix, além de serem práticos de se instalar. A versão do Postgrey, 1.32, foi escolhida por ser a mais atual para a versão do Mandriva utilizada. Já a versão do SPF, 2.007, foi selecionada por ser a mais atual disponível no site do projeto do SPF⁵.

O software anti-spam DSPAM foi escolhido por ser de fácil instalação e integração com o Postfix. Além disso, ele possui uma interface web que pode ser acessada pelos usuários para realizar o treinamento do filtro, aumentando assim, o nível de acertos na classificação das mensagens. Além disso, o DSPAM se integra facilmente ao software de antivírus ClamAV para realizar verificações de vírus nas mensagens. As versões dos softwares DSPAM e ClamAV - 3.8.0 e 0.95.1, respectivamente - foram escolhidas por serem as mais atuais para o Mandriva 2009.0.

A ferramenta Postfix Admin foi escolhida por possuir uma interface limpa e fácil de utilizar. Outro motivo para sua adoção foi sua fácil integração com o banco de dados PostgreSQL, local onde são armazenadas as contas de e-mail do servidor.

5 Maiores informações sobre o projeto SPF em: <http://www.openspf.org/Project_Overview>. Acesso em 26 mai 2009.

A escolha do software de webmail Round Cube foi definida pelo fato de sua instalação ser simples, além de o mesmo possuir uma interface limpa e fácil de se utilizar. A versão 0.2.1 foi escolhida por ser a mais atual disponível no site do software.

O principal critério para a escolha do banco de dados PostgreSQL foi a experiência de uso com o software. Além disso, ele é suportado por diversas ferramentas e sua integração é bastante simples. A versão utilizada, 8.3, foi escolhida por ser a mais atual para a distribuição Mandriva 2009.0.

Outro software cujo critério de adoção está relacionado à experiência de uso foi o servidor web Apache. A versão utilizada, 2.2.9 foi utilizada por ser a mais atual disponibilizada pela versão do Mandriva utilizada.

4.2. Implementação manual do servidor

Antes de se iniciar o processo de remasterização da imagem de instalação sistema, foi realizada a implementação do servidor de forma manual. Grande parte das configurações realizadas durante essa etapa foi feita sem a ajuda de scripts ou modelos de arquivos de configuração.

4.2.1. Cenário proposto

Para a elaboração do cenário foi utilizado o software VMware Server, na versão 2.0.1. Através desse software foi criada uma sub-rede com endereço IP 192.168.37.0/24. Todas as máquinas virtuais possuem acesso à internet. A máquina real, onde se encontra instalado o software foi configurada com o IP 192.168.37.1. A partir daí, foram decididas as seguintes configurações para máquina virtual que atuará como o servidor:

1. IP do servidor: 192.168.37.10;
2. FQDN⁶ do servidor: server.projetofinal.org;
3. Servidor DNS utilizado: 192.168.37.15.

O terceiro item citado acima diz respeito a um servidor auxiliar - que será utilizado pelo servidor de e-mails e pela máquina real - criado com a finalidade de atender as solicitações de resolução de nomes pelo mesmo. Para isso foi configurado no servidor auxiliar o software BIND⁷, que é um servidor DNS (*Domain Name System*) utilizado em sistemas GNU/Linux. Isso é necessário porque, para que a solução funcione corretamente, são necessárias configurações de nomes relacionadas ao domínio do servidor que apontam para o IP 192.168.37.10. O modo como foi configurado o serviço DNS não será abordado nesse projeto, visto que o

6 Fully Qualified Domain Name. Maiores informações em: <<http://www.linfo.org/fqdn.html>>. Acesso em 7 mai 2009.

7 Maiores informações e download em: <<https://www.isc.org/software/bind>>. Acesso em 7 mai. 2009.

projeto não contempla a configuração de servidores DNS. As configurações realizadas no servidor auxiliar foram:

1. Criação da zona DNS **projetofinal.org**;
2. Criação dos registros CNAME **admin.projetofinal.org**, **webmail.projetofinal.org** e **mail.projetofinal.org** apontando para o IP do servidor de e-mails: **192.168.37.10**;

A partir dessas configurações a implementação foi iniciada. Todas as configurações foram realizadas a partir da máquina real através do terminal de acesso do VMWare Server. Os resultados foram testados também a partir da máquina real, realizando requisições no servidor de e-mails. A instalação da distribuição Mandriva Linux 2009.0 no servidor de e-mail será abordada no próximo tópico.

A partir de figura 12 é possível verificar as configurações citadas acima, além de visualizar o cenário proposto:

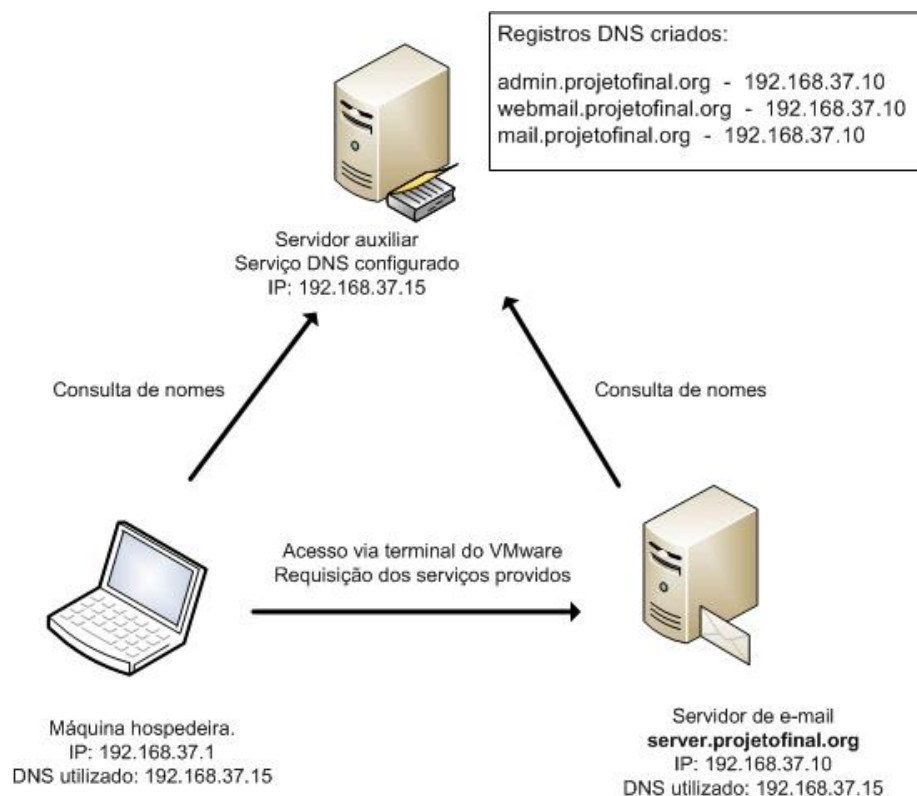


Figura 12: "Cenário proposto para a implementação".

Antes de se iniciar a descrição dos passos executados para a instalação e configuração dos softwares utilizados neste projeto, é importante observar as notações utilizadas.

Como já foi dito anteriormente, toda a configuração foi realizada a partir da máquina real, acessando o servidor de e-mail através do terminal de acesso do software VMware Server. Todas as configurações foram realizadas com o usuário root, que é o administrador do sistema e possui todas as permissões necessárias para a configuração dos serviços implementados no projeto.

Após a conexão remota no servidor de e-mail, é exibido um prompt do comando, onde os comandos podem ser executados. O prompt da distribuição Mandriva Linux é o BASH (*Bourne Again Shell*). O BASH é amplamente utilizado em distribuições GNU/Linux, por ser bastante robusto e incorporar características de outros famosos prompts de comando Shell⁸, como por exemplo, o KSH (Korn Shell).

As linhas de comando executadas possuem o seguinte formato:

```
# cp teste.txt teste-final.txt
```

A linha acima descreve um comando (`cp teste.txt teste-final.txt`) sendo executado como usuário root, devido ao símbolo # no início da linha. Já a exibição dos arquivos de configuração utiliza o seguinte formato: caminho do arquivo de configuração em negrito, seguido do conteúdo do arquivo. Exemplo:

/etc/hosts:

```
192.168.1.12      servidor servidor.intra
192.168.1.13      ns1 ns1.intra
```

⁸ Nome utilizado para designar interface com o usuário. Um exemplo famoso de Shell é o BASH, amplamente utilizado em distribuições GNU/Linux.

4.2.2. Instalação do Mandriva Linux 2009.0

Para a instalação do Mandriva Linux, versão 2009.0, foi utilizada a imagem de instalação original da distribuição. Essa imagem pode ser baixada direto do site do Mandriva ou através de torrents⁹.

A partir da imagem de instalação do Mandriva Linux 2009.0, baixada do site oficial do Mandriva, foi iniciada a instalação do sistema no servidor. Para a instalação, foi criada uma máquina virtual no software VMware Server. Na configuração dessa máquina virtual, foi escolhida a opção de se utilizar uma imagem ISO 9660 no carregamento inicial. O formato ISO 9660 é utilizado pela imagem de instalação do Mandriva. Com isso, o software irá emular a situação de se inserir um cd de instalação do sistema na máquina virtual. A figura 13 mostra a opção de utilizar uma imagem ISO 9660 durante a criação de uma máquina virtual no software VMware Server:

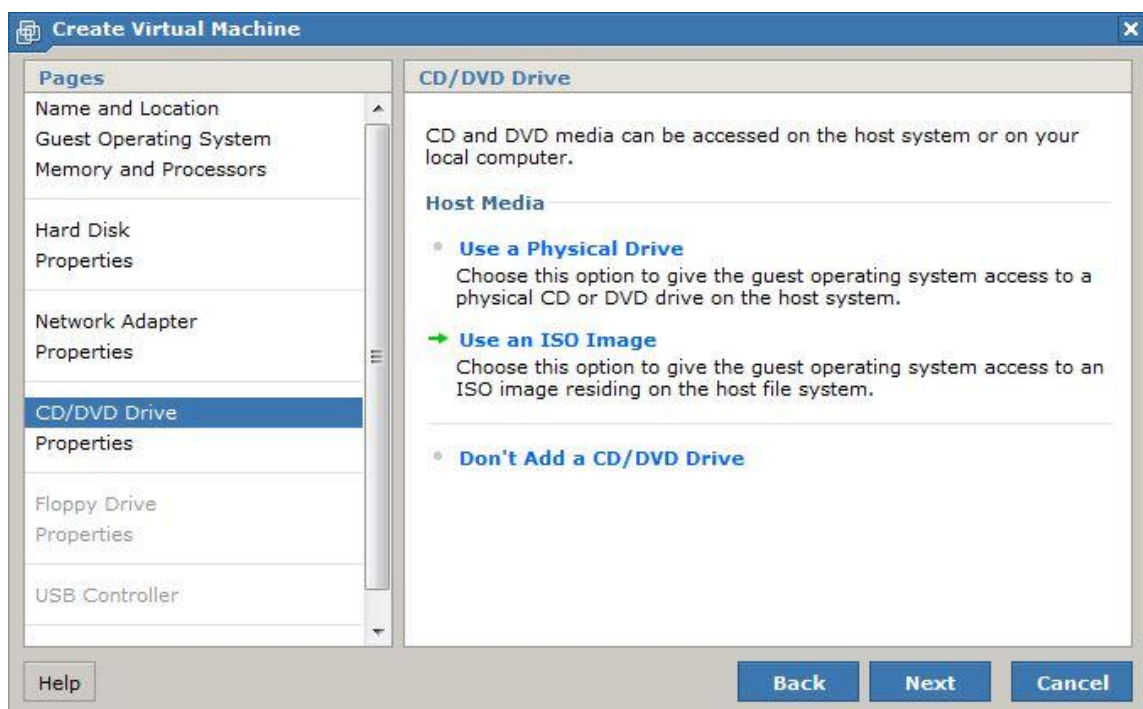


Figura 13: “Opção de utilizar uma imagem ISO 9660 no software VMware Server.”

⁹ Arquivos utilizados por clientes BitTorrent. Maiores informações em:
http://www.bittorrent.org/beps/bep_0003.html >. Acesso em 16 mai 2009.

Após o término da criação da máquina virtual, a mesma foi iniciada. A partir desse momento inicia-se o processo de instalação do sistema. A primeira tela exibida mostra uma série de opções, entre elas, existe a opção de se instalar o Mandriva. Existem também as opções:

- Carregar um sistema do disco rígido;
- Entrar no sistema em modo de recuperação, útil para a resolução de problemas com o sistema;
- Realizar um teste de memória na máquina.

A partir de figura 14 é possível verificar a existência da opção de instalar o Mandriva, na tela inicial de instalação do sistema.



Figura 14: “Tela inicial de instalação do Mandriva Linux.”

Após escolhida a opção de instalar o sistema foi exibida uma tela solicitando a escolha do idioma. A partir dessa escolha o sistema instala os pacotes necessários para a exibição dos caracteres de acordo com o idioma escolhido, além de configurar o teclado de acordo com o padrão do país.

A próxima etapa da instalação trata do particionamento do servidor. Em sistemas GNU/Linux é possível segmentar a estrutura de diretórios em diversas partições. A vantagem disso é prevenir o servidor de um travamento caso um ou mais desses diretórios cresça de forma incontrolável na ocorrência de um erro no sistema. Para a implementação do projeto, o sistema foi segmentado em 10 partições. O quadro 4 mostra as partições criadas e suas respectivas funções:

Quadro 4: “Esquema de partições criado para o projeto.”

Partição	Descrição
/boot	Diretório onde se encontram os arquivos do kernel do sistema, que são responsáveis por detectar todo o hardware da máquina.
/tmp	Diretório com a finalidade de guardar arquivos temporários.
/usr	Diretório onde ficam arquivados os softwares instalados no sistema.
/home	Diretório onde ficam os arquivos pessoais dos usuários. Será neste local onde serão armazenadas todas as caixas postais das contas de e-mail criadas.
/var	Dados variáveis do sistema. Neste diretório ficam os arquivos de log do sistema.
/var/spool/deleted-maildirs	Diretório utilizado pela ferramenta Postfix Admin para armazenar as contas de e-mail que foram removidas. A instalação e configuração dessa ferramenta será abordada no tópico 4.2.3.1.
/var/lib/pgsql	Diretório onde ficam armazenados os dados do banco de dados PostgreSQL.
/var/www	Diretório base para os sites criados no servidor web Apache.
swap	Área de troca utilizada pelo sistema. Geralmente é utilizada quando toda a memória RAM está sendo utilizada.

Após criado o esquema de partições, foi escolhido o tipo sistema de arquivos a ser configurado nas partições. O XFS foi escolhido por ser um sistema sensivelmente mais rápido na recuperação em caso de uma interrupção por falta de energia, por exemplo.

Segundo a *Silicon Graphics*, o XFS possui uma estrutura de recuperação que permite a reinicialização rápida do sistema após uma interrupção inesperada, diferente de outros sistemas de arquivos – como o Ext3, por exemplo – que necessitam de verificações no sistema de arquivos antes da reinicialização, o que pode levar horas para finalizar. [SILICOM GRAPHICS, 2006]

A figura 15 mostra o menu de particionamento do instalador do Mandriva Linux, com todas as partições criadas para o projeto:

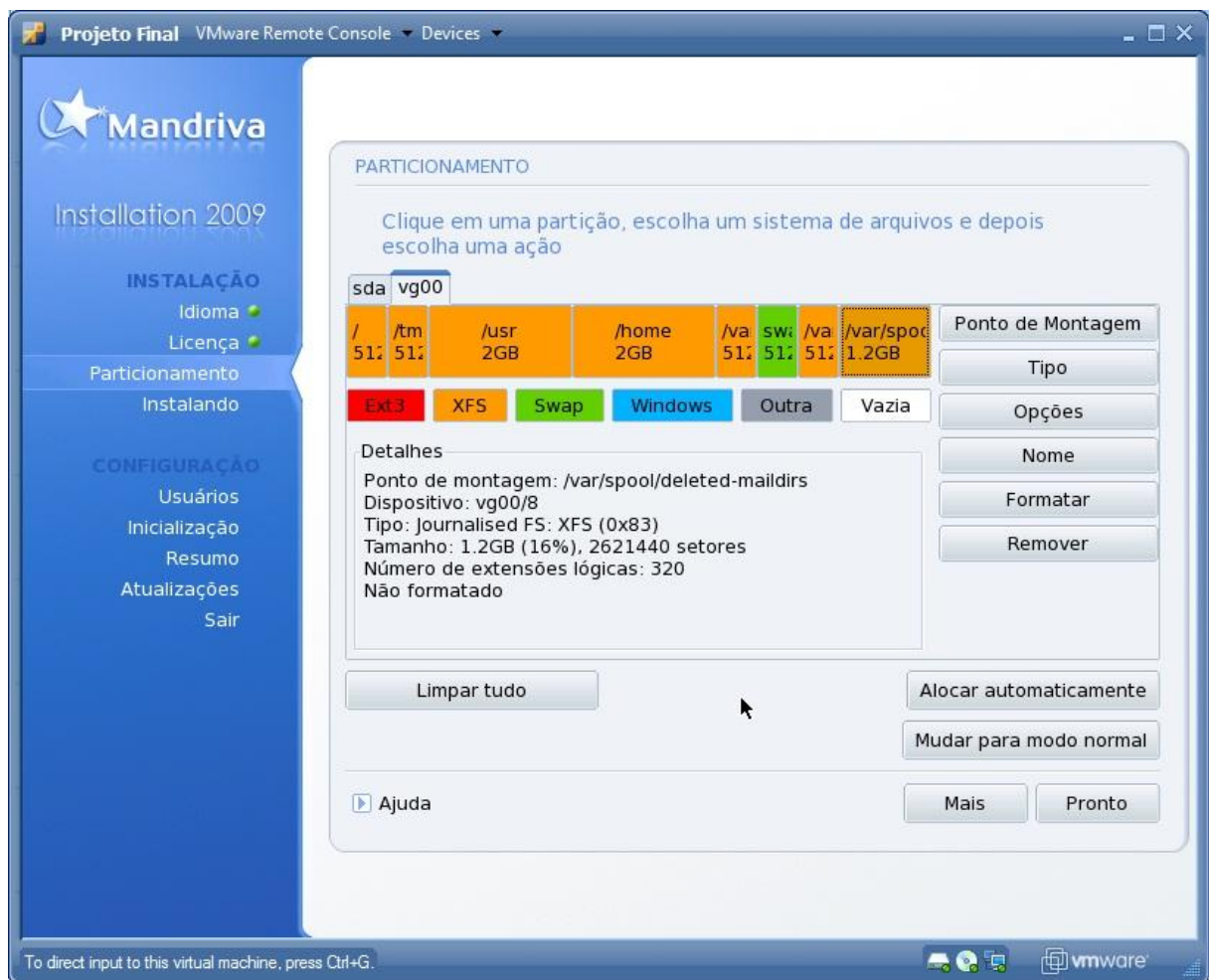


Figura 15: “Menu de particionamento do instalador do Mandriva Linux.”

Após o particionamento do servidor, foi exibido um menu para seleção de pacotes a serem instalados. Como a idéia do projeto é instalar somente o necessário, descartando softwares que não contribuem para o funcionamento da solução, foram desmarcadas todas as opções mostradas. Com isso, softwares desnecessários – como suíte de escritório, jogos e etc. – foram excluídos da instalação, restando somente os programas necessários para o funcionamento básico do sistema. A partir de figura 16, é possível verificar as todas as opções desmarcadas na tela de seleção de pacotes:

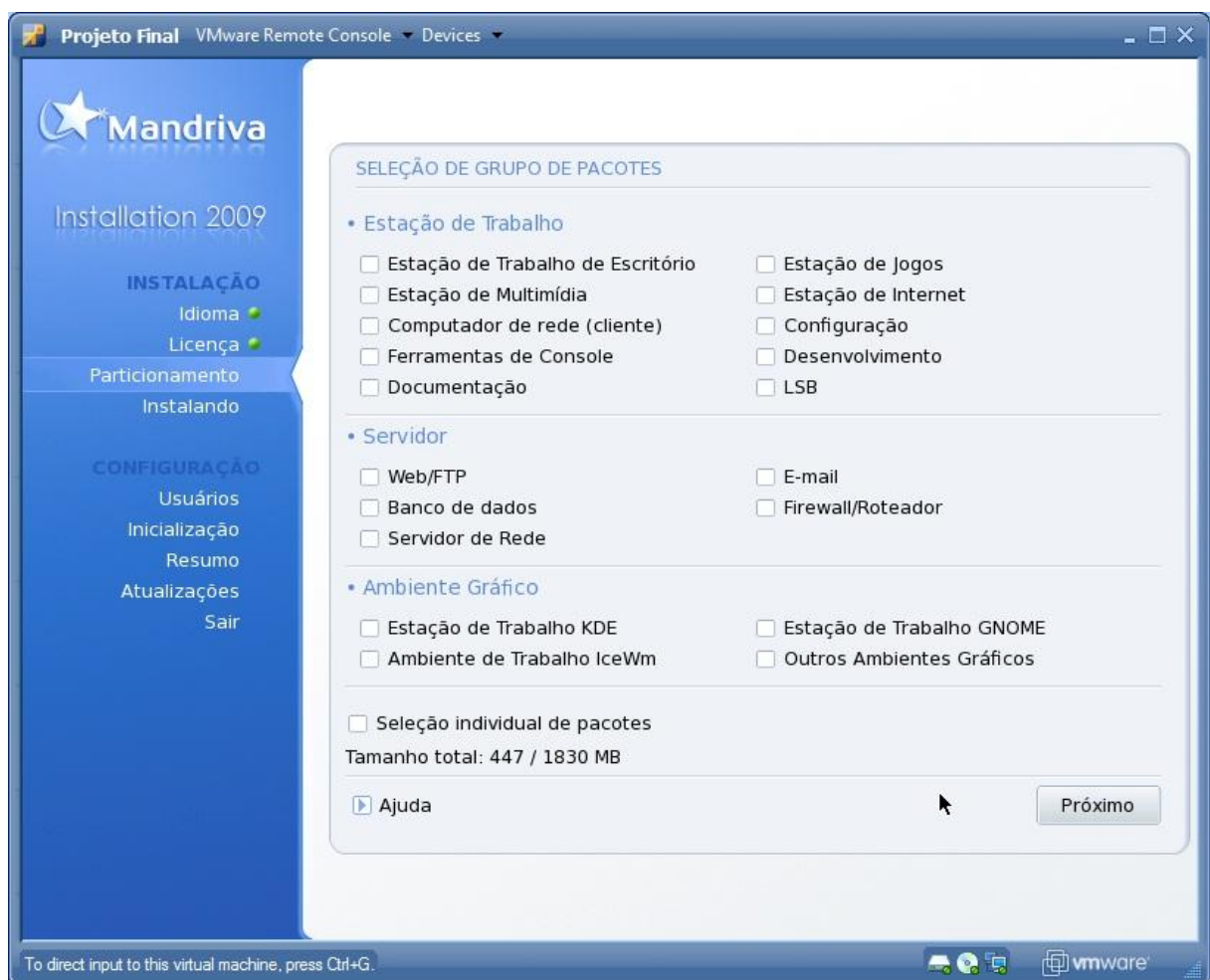


Figura 16: “Tela de seleção de pacotes do instalador do Mandriva Linux.”

O próximo passo da instalação tratou da criação de usuários do sistema, além da criação da senha do usuário root, que possui permissões administrativas no servidor. Nessa etapa foi criado o usuário comum projeto, além de ser configurada a senha para o usuário root.

Depois de criado o usuário projeto e configurada a senha do usuário root, o instalador do Mandriva Linux exibiu uma tela informando que a instalação do sistema foi concluída com sucesso. O sistema foi reiniciado depois disso, dando início a implementação do servidor de e-mail.

4.2.3. Implementação do servidor de e-mail

Este tópico aborda a implementação do servidor de e-mail. Durante esse tópico não serão abordadas as configurações de segurança, visto que esse assunto será dissertado no tópico 4.2.4.

4.2.3.1. Configuração do banco de dados e do Postfix Admin

Antes de iniciar a configuração dos serviços responsáveis pela troca de e-mails, foi instalado o banco de dados responsável por armazenar todas as informações dos usuários de e-mail do sistema. O software em questão é o PostgreSQL. Ele foi instalado a partir do seguinte comando:

```
# urpmi postgresql8.3-server
```

Depois disso, foi realizada a criação da base de dados onde irão ficar as informações dos usuários do servidor. O nome da base de dados foi denominado **postfix**. Para a criação dessa base de dados foi utilizado o prompt de comando do PostgreSQL. Por questões de segurança, foi criado um usuário no banco de dados, chamado postfix, que possui acesso total apenas à base de dados postfix. Depois de feita a criação do usuário, foi criada a base de dados. A partir da figura 17, pode-se observar os comandos para a criação do usuário postfix, além da base de dados postfix, no prompt de comando do PostgreSQL:


```
[root@server rpms]# psql -U postgres
Bem vindo ao psql 8.2.10, o terminal iterativo do PostgreSQL.

Digite: \copyright para mostrar termos de distribuição
        \h para ajuda com comandos SQL
        \? para ajuda com comandos do psql
        \g ou terminar com ponto-e-vírgula para executar a consulta
        \q para sair

postgres=# CREATE USER postfix WITH PASSWORD '3ed4rf';
CREATE ROLE
postgres=# CREATE DATABASE postfix OWNER postfix;
CREATE DATABASE
postgres=#
```

Figura 17: “Criação da base de dados postfix no banco de dados.”

Com a base de dados dos usuários criada, o próximo passo da implementação do projeto foi a instalação da ferramenta Postfix Admin. A versão do Postfix Admin baixada – o porquê da versão já foi explicado no tópico 4.1 – foi descompactada dentro do diretório **/var/www**. Após descompactação, os arquivos do Postfix Admin ficaram situados dentro de um diretório chamado **postfixadmin**. A figura 18 mostra os arquivos do software Postfix Admin, situados no diretório **/var/www/postfixadmin**:

```
[root@server postfixadmin]# pwd
/var/www/postfixadmin
[root@server postfixadmin]# ls
ADDITIONS/          edit-active-domain.php  list-virtual.php
admin/              edit-active.php         login.php
backup.php          edit-admin.php          logout.php
broadcast-message.php edit-alias.php           main.php
CHANGELOG.TXT       edit-domain.php         model/
common.php          edit-mailbox.php        password.php
config.inc.php      edit-vacation.php       search.php
create-admin.php    fetchmail.php           sendmail.php
create-alias-domain.php functions.inc.php        setup.php
create-alias.php    GPL-LICENSE.TXT         templates/
create-domain.php   images/                 tests/
create-mailbox.php  index.php               upgrade.php
css/               INSTALL.TXT             users/
debian/            languages/              variables.inc.php
delete.php         LICENSE.TXT             viewlog.php
DOCUMENTS/         list-admin.php          VIRTUAL_VACATION/
edit-active-admin.php list-domain.php          xmlrpc.php
[root@server postfixadmin]#
```

Figura 18: “Arquivos do software Postfix Admin.”

Ainda acerca da figura 18 é possível perceber a presença dos arquivos **config.inc.php** e **setup.php**. O primeiro trata da configuração do Postfix Admin. Já

o segundo é um script que cria as tabelas necessárias para o funcionamento do software. O seguinte trecho abaixo trata da configuração da base de dados criada no servidor de banco de dados PostgreSQL:

```
/var/www/postfixadmin/config.inc.php:
// Database Config
// mysql = MySQL 3.23 and 4.0, 4.1 or 5
// mysqli = MySQL 4.1+
// pgsql = PostgreSQL
$CONF['database_type'] = 'pgsql';
$CONF['database_host'] = 'localhost';
$CONF['database_user'] = 'postfix';
$CONF['database_password'] = '3ed4rf';
$CONF['database_name'] = 'postfix';
```

As linhas iniciadas por duas barras (//) indicam comentários. Os parâmetros **database_type** e **database_host** configuram o banco de dados utilizado e o endereço onde se encontra o banco de dados, respectivamente. Já os parâmetros **database_user**, **database_password** e **database_name** configuram, respectivamente: o usuário dono da base de dados, a senha do usuário dono da base de dados e o nome da base de dados. Todas as configurações foram feitas de acordo com os parâmetros utilizados na criação da base de dados postfix.

Após a configuração do arquivo config.inc.php, o próximo passo seria executar o script setup.php via browser. Porém, para isso é necessário configurar um servidor web com a finalidade de conseguir executar esse arquivo via browser.

Para a configuração do servidor web, foi instalado o software Apache. Foi necessário também a instalação do PHP5, ferramenta indispensável ao funcionamento do Postfix Admin. Para a instalação desses softwares, foi executado o seguinte comando:

```
# urpmi apache-base apache-mod_php
```

O servidor web apache possui um recurso bastante utilizado que é chamado de *Virtual Hosts*. Através desse recurso, é possível criar diversas páginas web, com configurações distintas, em um mesmo servidor. Em sistemas Mandriva Linux, o diretório onde ficam os arquivos de configuração de *Virtual Hosts* é o **/etc/httpd/conf/vhosts.d**. O Apache irá carregar todos os arquivos que estiverem dentro desse diretório e possuam em seu nome o sufixo **.conf**. Diante disso, foi criado o arquivo **postfixadmin.conf**, dentro de **/etc/httpd/conf/vhosts.d**, com o seguinte conteúdo:

/etc/httpd/conf/vhosts.d/postfixadmin.conf:

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerAdmin webmaster@postfixadmin
    DocumentRoot /var/www/postfixadmin
    ServerName admin.projetoefinal.org
    ErrorLog /var/log/httpd/postfixadmin-error_log
    CustomLog /var/log/httpd/postfixadmin-access_log common
</VirtualHost>
```

Os parâmetros que merecem atenção são:

- **NameVirtualHost:** Indica em que endereço IP e em qual porta o VirtualHost está aguardando por requisições. No exemplo **NameVirtualHost *:80**, o VirtualHost está aguardando em qualquer IP configurado no servidor, desde que seja na porta 80/TCP;
- **DocumentRoot:** local no servidor onde estão localizados os arquivos da página web. Também é chamado de diretório web;
- **ServerName:** parâmetro mais importante. Define o nome do servidor virtual. Para acessar a página definida por esse *Virtual Host*, o browser deve acessar a url indicada nesse parâmetro. Exemplo: caso o ServerName de um Virtual Host **teste.exemplo.com.br** e seu diretório web é **/var/www/exemplo.com.br**, para acessar o site que está dentro

desse diretório, é necessário acessar a url **teste.exemplo.com.br** em um navegador de internet;

- **ErrorLog**: local onde será gerado o arquivo de log de erros do *Virtual Host*.

Depois de realizada a configuração do Virtual Host **admin.projetofinal.org** no Apache, foi acessado o endereço **http://admin.projetofinal.org/setup.php** através de um navegador na máquina real. Ao acessar o endereço citado anteriormente, o Postfix Admin cria a estrutura de tabelas necessária e solicita a criação de um usuário administrador da ferramenta. O usuário criado foi **admin@projetofinal.org**. A partir da figura 19 podemos verificar a criação das tabelas pelo software Postfix Admin e a pedido de criação de um usuário que irá administrar a ferramenta:

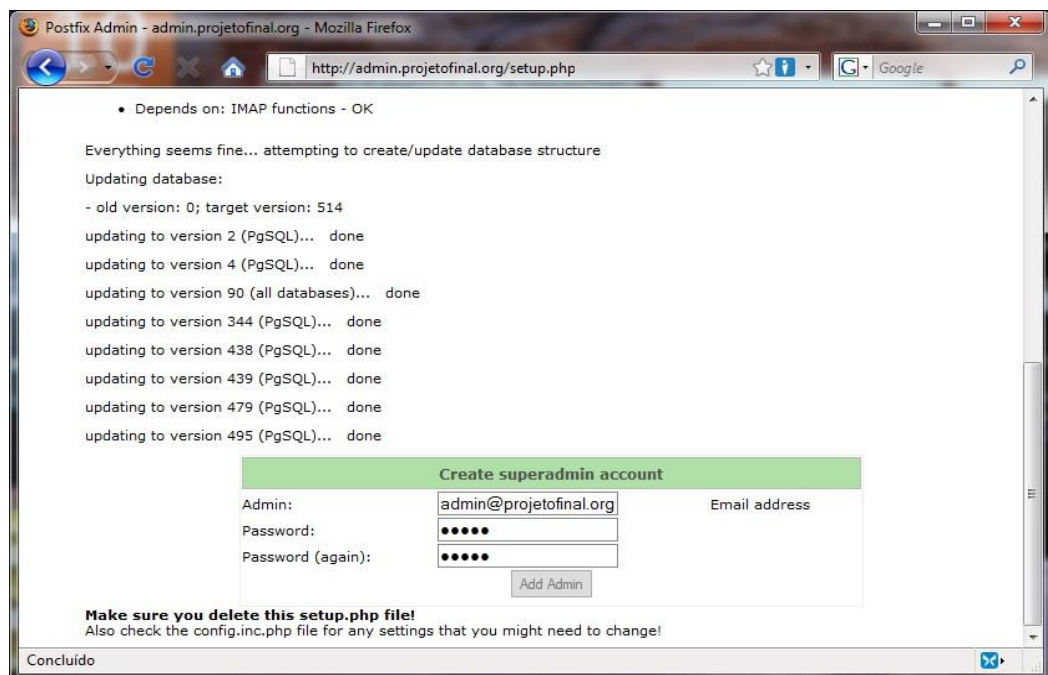
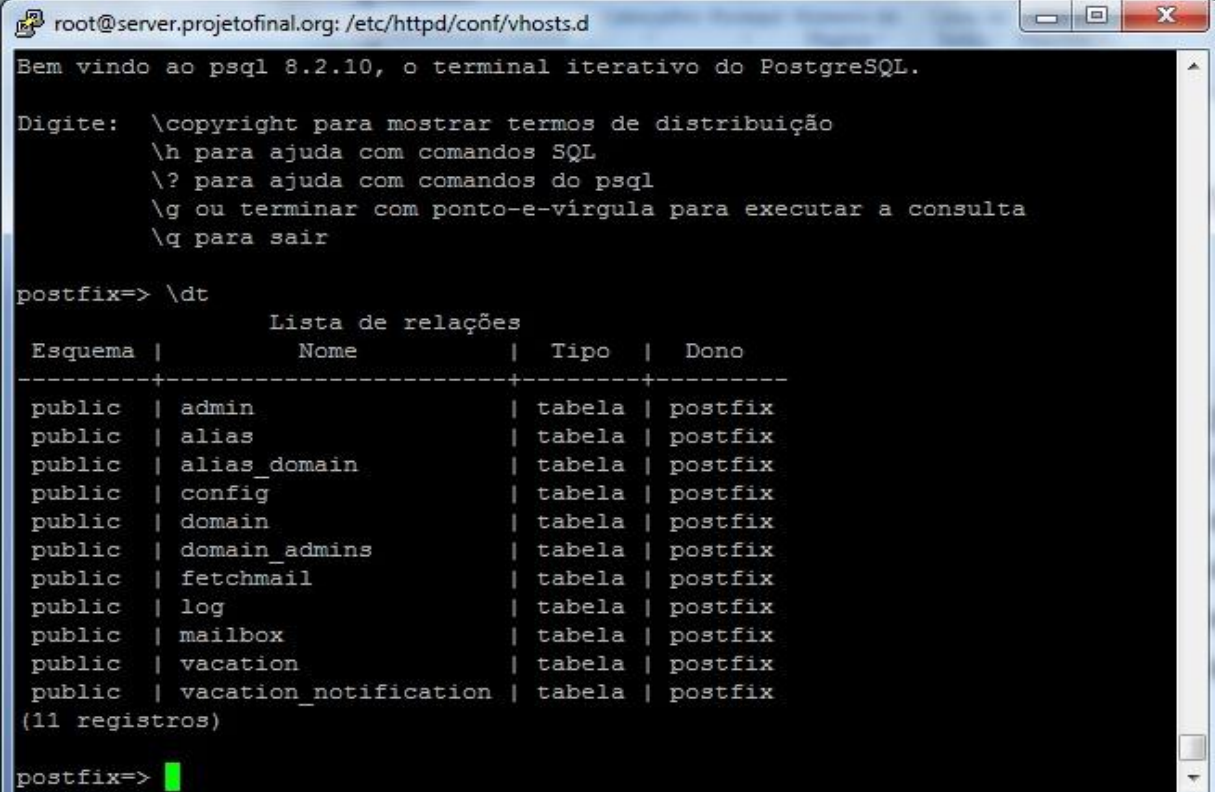


Figura 19: “Criação das tabelas pelo Postfix Admin.”

É importante lembrar que, devido as configurações de DNS do servidor auxiliar, o host **admin.projetofinal.org** aponta para o IP 192.168.37.10, pertencente ao servidor de e-mails. A partir desse fato é possível perceber a importância do servidor auxiliar na implementação do projeto, pois caso a configuração do DNS não estivesse realizada não seria possível executar o script **setup.php** através de um

navegador de internet. A figura 20 mostra a estrutura de tabelas criada pelo Postfix Admin. Essa estrutura será a base de dados de autenticação utilizada pelos softwares Postfix e Dovecot, cuja instalação e configuração será descrita no próximo tópico.



```

root@server.projetofinal.org: /etc/httpd/conf/vhosts.d
Bem vindo ao psql 8.2.10, o terminal iterativo do PostgreSQL.

Digite: \copyright para mostrar termos de distribuição
        \h para ajuda com comandos SQL
        \? para ajuda com comandos do psql
        \g ou terminar com ponto-e-vírgula para executar a consulta
        \q para sair

postfix=> \dt
                Lista de relações
Esquema |          Nome          | Tipo |  Dono
-----+-----+-----+-----
public  | admin                  | tabela | postfix
public  | alias                  | tabela | postfix
public  | alias_domain           | tabela | postfix
public  | config                 | tabela | postfix
public  | domain                 | tabela | postfix
public  | domain_admins          | tabela | postfix
public  | fetchmail              | tabela | postfix
public  | log                    | tabela | postfix
public  | mailbox                | tabela | postfix
public  | vacation               | tabela | postfix
public  | vacation_notification | tabela | postfix
(11 registros)

postfix=>

```

Figura 20: “Estrutura de tabelas criada pelo Postfix Admin.”

Após a instalação do Postfix Admin, foi acessado o endereço <http://admin.projetofinal.org> e efetuado o *login* com as credenciais configuradas durante a execução do script **setup.php**. A partir da figura 21, é possível observar interface de administração do Postfix Admin:

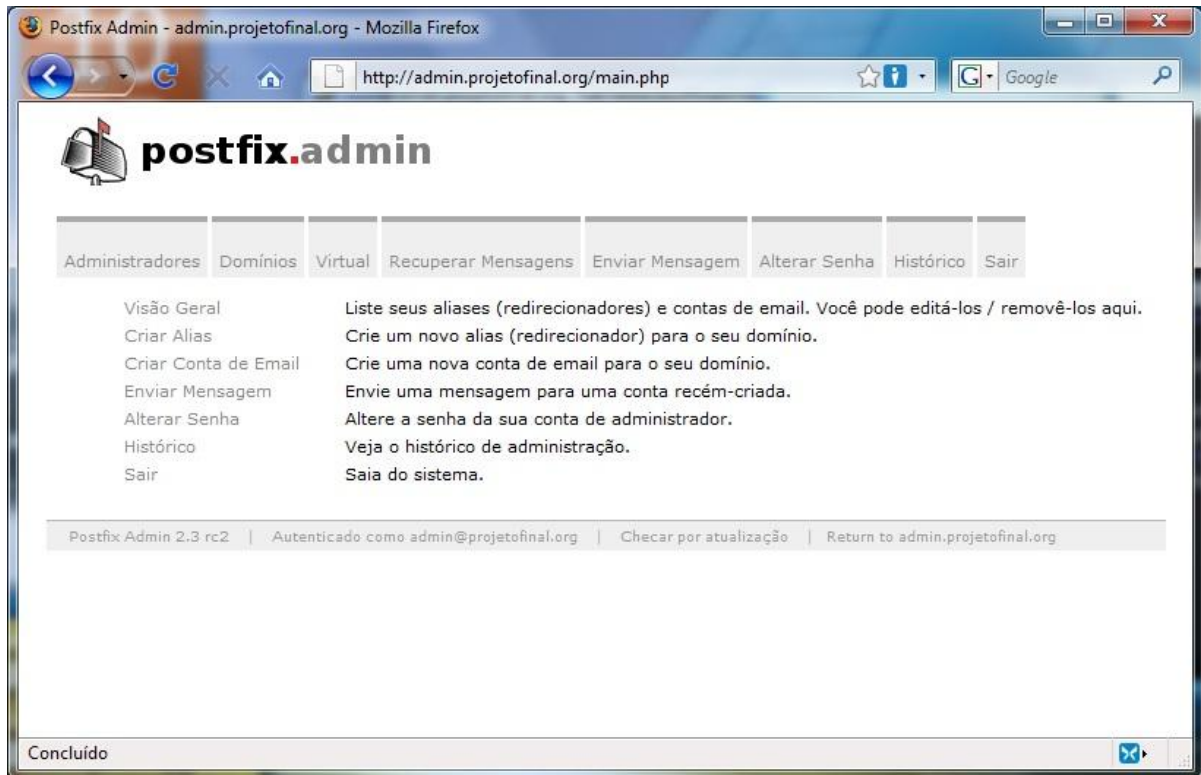


Figura 21: “Interface de administração do Postfix Admin.”

4.2.3.1. Configuração dos serviços de e-mail e do webmail

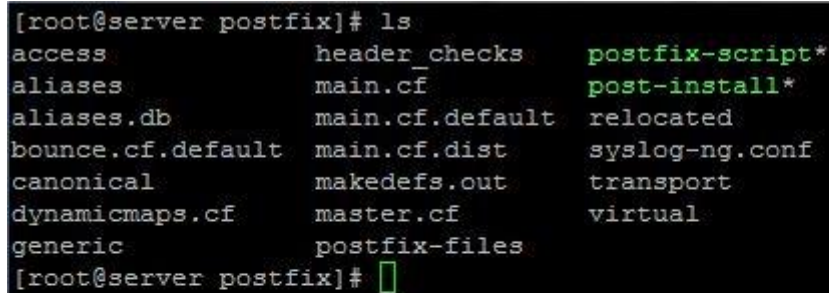
Depois de configurado a base de dados do servidor e a ferramenta que irá administrar as contas de e-mail, foi necessária a instalação dos softwares Postfix e Dovecot. O primeiro possui a função de um MTA, ou seja, ele é o servidor de e-mail propriamente dito. Seu papel é enviar e receber as mensagens através do protocolo SMTP. Já o segundo tem a função de gravar as mensagens na caixa postal correspondente, além de aguardar por requisições de usuários para realizar a entrega da mensagem, através dos protocolos POP e IMAP. Geralmente a mensagem é entregue para um MUA, que é um cliente de e-mail, como por exemplo, o Thunderbird¹⁰. Existem outras funcionalidades do software Dovecot que serão utilizadas no projeto. Elas serão descritas no tópico 4.2.3.3.

Para a instalação do Postfix e do Dovecot, foi executado o seguinte comando:

```
# urpmi postfix dovecot
```

¹⁰ Cliente de e-mail da fundação Mozilla. Download disponível em: <<http://www.mozillamessaging.com/en-US/>>. Acesso em: 13 mai 2009.

Feita a instalação dos softwares, foi iniciada a configuração do Postfix. Seu diretório de configuração é o /etc/postfix. Neste diretório existem arquivos que configuram os diversos recursos que o Postfix possui. A figura 22 mostra os arquivos de configuração do postfix, dentro do diretório /etc/postfix:



```
[root@server postfix]# ls
access          header_checks   postfix-script*
aliases         main.cf         post-install*
aliases.db      main.cf.default relocated
bounce.cf.default main.cf.dist    syslog-ng.conf
canonical       makedefs.out   transport
dynamicmaps.cf master.cf       virtual
generic         postfix-files
[root@server postfix]#
```

Figura 22: “Arquivos de configuração do postfix.”

O Postfix pode trabalhar com três tipos distintos de transporte:

- **Local:** o Postfix é configurado para trabalhar com um domínio local, o que significa que a base de dados dos usuários deverá estar no sistema. Com esse tipo de configuração é possível administrar apenas um domínio no servidor; [DENT, 2003]
- **Relay:** faz com que o Postfix atue como um portão de entrada para os e-mails. Após o recebimento da mensagem, ela é entregue para outro servidor de e-mails, geralmente na mesma rede; [DENT, 2003]
- **Virtual:** configura o Postfix para utilizar domínios virtuais, onde a base de dados dos usuários se encontra em um local diferente da base de dados do sistema, como por exemplo, um banco de dados. Esse tipo

de transporte permite a configuração de mais de um domínio em um mesmo servidor. [DENT, 2003]

O transporte virtual foi escolhido para a implementação do projeto, devido à possibilidade de se configurar diversos domínios em um mesmo servidor.

O arquivo de configuração principal do Postfix é o **main.cf**. Os principais parâmetros configurados foram:

/etc/postfix/main.cf:

```
mydomain = projetofinal.org
myhostname = server.projetofinal.org
virtual_transport = virtual
virtual_mailbox_base = /home/vmail
virtual_mailbox_domains = pgsql:/etc/postfix/pgsql/virtual-domains.cf
virtual_mailbox_maps = pgsql:/etc/postfix/pgsql/virtual-mailbox-maps.cf
```

O parâmetro **mydomain** configura o domínio do servidor. Como o tipo de transporte utilizado na implementação será o virtual, esse parâmetro indica o domínio principal. Já o parâmetro **myhostname** indica o FQDN do servidor. Os parâmetros restantes estão relacionados à configuração do Postfix para o tipo de transporte virtual e são descritos a seguir:

- **virtual_transport**: Indica qual é o tipo de transporte utilizado pelo postfix;
- **virtual_mailbox_base**: Diretório base onde fica as caixas postais do usuários. O Postfix utiliza o formato **<base>/domínio/usuário** para guardar as mensagens. Por exemplo: no caso da implementação do projeto, as caixas postais dos usuários serão guardadas no formato **/home/vmail/domínio/usuário**;

- **virtual_mailbox_domain**: Indica o caminho para o arquivo de configuração que possui os parâmetros necessários para pesquisar por domínios cadastrados na base de dados;
- **virtual_mailbox_maps**: Indica o caminho para o arquivo de configuração que possui os parâmetros necessários para pesquisar por contas de e-mail cadastradas na base de dados;

É importante atentar para formato dos parâmetros **virtual_mailbox_domain** e **virtual_mailbox_maps**. Esses parâmetros são configurados seguindo o seguinte formato: **<base de dados>:<caminho para o arquivo de configuração>**. No caso da implementação do projeto, a primeira parte do parâmetro, **pgsql**, indica que a base de dados a ser utilizada pelo Postfix será o banco de dados PostgreSQL, cuja configuração já foi abordada no tópico anterior. A partir da figura 23, pode-se visualizar o conteúdo do arquivo **/etc/postfix/pgsql/virtual-mailbox-maps.cf**, que possui os parâmetros **user**, **password**, **dbname** e **hosts**, utilizados para o Postfix conseguir conectar-se na base de dados, além do parâmetro **query**, que permite ao postfix realizar a pesquisa por contas de e-mail.

```
[root@server pgsql]# cat virtual-mailbox-maps.cf
user      = postfix
password  = 3ed4rf
dbname    = postfix
hosts     = localhost
query     = SELECT maildir FROM mailbox WHERE username='%s' AND active = true
```

Figura 23: “Parâmetros do arquivo virtual-mailbox-maps.cf.”

Além das configurações básicas realizadas, foi habilitado no Postfix o sistema de quotas de e-mail. Elas definem o tamanho máximo que a caixa postal de uma conta de e-mail pode possuir. Os valores das quotas são armazenados na mesma base de dados onde se encontram os domínios e as contas de e-mail. O seguinte trecho – inserido no arquivo **main.cf** - indica a configuração das quotas no servidor do projeto:

/etc/postfix/main.cf:

```
virtual_mailbox_limit_maps = pgsql:/etc/postfix/pgsql/virtual-mailbox-limit-maps.cf
```



```
virtual_overquota_bounce = Yes
```

```
virtual_maildir_limit_message = "Conta de e-mail com limite de quota ultrapassado."
```

O parâmetro **virtual_mailbox_limit_maps** indica o caminho para o arquivo de configuração que possui os parâmetros necessários para pesquisar pelas quotas das contas de e-mail cadastradas na base de dados. Já o parâmetro **virtual_overquota_bounce** indica se o Postfix deverá retornar uma mensagem para o remetente caso o destinatário esteja com sua quota acima do limite estabelecido. Por fim, o parâmetro **virtual_maildir_limit_message** configura a mensagem de erro que deverá ser enviada para o remetente.

A partir desse momento, o servidor de e-mail já é capaz de enviar ou receber e-mails. Porém, existe uma limitação: a capacidade do servidor de enviar ou receber e-mails limita-se à rede especificada no cenário proposto, tópico 4.2.1. Isso ocorre devido a uma série de fatos, dos quais é possível destacar:

1. O domínio projetofinal.org até o momento é fictício. Ele não foi cadastrado nos serviços de registros de domínios na internet, como por exemplo, o RegistroBR¹¹;
2. É necessário um gateway com IP fixo na internet para que o servidor possa atender as requisições de servidores localizados de internet. As máquinas virtuais possuem acesso à internet, porém através de um gateway com IP dinâmico.

Depois de configurado o Postfix, foi iniciada a configuração do Dovecot. Seu principal arquivo de configuração é o **dovecot.conf** situado dentro do diretório **/etc**. Os principais parâmetros configurados foram:

/etc/dovecot.conf:

```
protocols = imap pop3
```

```
mail_location = maildir:/home/vmail/%d/%n
```

```
auth default {
```

¹¹ Registro de domínios para a internet no Brasil. Maiores informações em: <<http://registro.br/index.html>>. Acesso em 13 mai 2009.

```

passdb sql {
    args = /etc/dovecot-pgsql.conf
}
userdb sql {
    args = /etc/dovecot-pgsql.conf
}
}

```

O parâmetro **protocols** especifica quais protocolos o Dovecot suporta. No momento, os protocolos habilitados são o IMAP e o POP. Existem outros dois, o IMAPS e o POPS, porém esses são discutidos mais adiante. O parâmetro **mail_location** indica qual o tipo das caixas postais – pode ser do tipo mbox ou maildir – e o diretório base onde ficam as caixas postais.

É importante observar o formato desse parâmetro. Ele segue o seguinte formato: **<tipo de caixa postal>:<diretório base das caixas postais>**. Como pode ser observado no trecho acima, retirado do arquivo `/etc/dovecot.conf`, na implementação o formato das caixas postais é o **maildir** e o diretório base das caixas postais se encontra em **/home/vmail**. Por fim, os parâmetros **passdb sql** e **userdb sql** configuram o dovecot para utilizar o banco de dados SQL para realizar a autenticação. Cada parâmetro possui um argumento, indicado por **args =**, que indica o caminho para o arquivo de configuração que possui os parâmetros necessários para o Dovecot pesquisar pelas contas de e-mail e suas respectivas senhas cadastradas na base de dados. No caso da implementação do projeto tanto os parâmetros **passdb sql** e **userdb sql** utilizam o mesmo arquivo de configuração:

/etc/dovecot-pgsql.conf:

```

driver = pgsql
connect = host=localhost dbname=postfix user=postfix password=3ed4rf
user_query = SELECT username FROM mailbox WHERE username='%u';
password_query = SELECT password FROM mailbox WHERE username='%u';

```

O parâmetro **driver** indica qual o tipo de base de dados utilizada. Já o parâmetro **connect** indica os dados necessários para a conexão na base de dados.

Por fim, os parâmetros **user_query** e **password_query** realizam a busca na base de dados pelas contas de e-mail e suas respectivas senhas.

Neste momento da implementação, já é possível acessar uma caixa postal via IMAP ou POP para buscar a mensagens no servidor. Para isso é necessário um agente de e-mail. Agentes bastante conhecidos são o Outlook, da empresa Microsoft; e o Thunderbird, da fundação Mozilla.

Além desses agentes citados existe outro tipo de agente: o *webmail*. A finalidade do *webmail* é prover acesso à caixa postal de uma conta de e-mail através de um navegador de internet. Através do *webmail* é possível realizar o envio de mensagens, além de visualizar os e-mails que chegaram à caixa postal. Geralmente existem outros recursos, como o armazenamento de contatos, porém isso varia de acordo com a empresa que provê o serviço. Alguns serviços de webmail famosos são o *Gmail*, da empresa Google; e o *Yahoo Mail*, da empresa Yahoo.

No servidor do projeto, foi instalado o software de webmail *RoundCube*, versão 0.2.1. Para o seu funcionamento, foi preciso criar uma base de dados no PostgreSQL. Para isso, foi criado no banco de dados o usuário roundcube, que possui acesso irrestrito somente a base de dados roundcubemail. Após a criação do usuário roundcube, foi criada a base de dados. A partir da figura 24, pode-se observar os comandos para a criação do usuário roundcube, além da base de dados roundcubemail, no prompt de comando do PostgreSQL.

```
[root@server ~]# psql -U postgres
Bem vindo ao psql 8.2.10, o terminal iterativo do PostgreSQL.

Digite: \copyright para mostrar termos de distribuição
        \h para ajuda com comandos SQL
        \? para ajuda com comandos do psql
        \g ou terminar com ponto-e-vírgula para executar a consulta
        \q para sair

postgres=# CREATE USER roundcube WITH PASSWORD '3ed4rf5tg';
CREATE ROLE
postgres=# CREATE DATABASE roundcubemail OWNER roundcube;
CREATE DATABASE
postgres=# █
```

Figura 24: "Criação da base de dados roundcubemail."

Após a criação da base de dados, o software foi descompactado dentro do diretório **/var/www**, assim como o Postfix Admin, ferramenta que já foi abordada no tópico anterior. Após a descompactação, arquivos do software ficaram situados dentro do diretório **/var/www/roundcubemail**. A figura 25 mostra os arquivos do software *RoundCube*:

```
[root@server roundcubemail]# pwd
/var/www/roundcubemail
[root@server roundcubemail]# ls
bin/          index.php  logs/       robots.txt  temp/
CHANGELOG    INSTALL   program/    skins/       UPGRADING
config/       LICENSE   README     SQL/
[root@server roundcubemail]#
```

Figura 25: "Arquivos do software RoundCube."

Ainda na figura 25, pode-se perceber a existência dos diretórios **config** e **SQL**. O primeiro possui os arquivos de configuração do software; já o segundo possui os scripts SQL necessários para criar a estrutura de tabelas necessárias dentro da base de dados configurada para utilização do software.

O diretório **config**, situado dentro de **/var/www/roundcubemail**, possui dois arquivos importantes: **db.inc.php** e **main.inc.php**. O primeiro configura o software para conectar-se na base de dados criada para sua utilização. Já o segundo possui as configurações principais do *RoundCube*. O arquivo **db.inc.php** foi configurado da seguinte forma:

/var/www/roundcubemail/config/db.inc.php:

```
$rcmail_config['db_dsnw'] = 'pgsql://roundcube:3ed4rf5tg@localhost/roundcubemail';
```

A linha de configuração obedece ao seguinte padrão:

```
'<tipo de base de dados>://<usuário>:<senha>@<servidor>/<base de dados>'
```

Onde:

- **Tipo de base de dados:** Especifica qual o tipo de base de dados a ser utilizada. Como o PostgreSQL é o banco de dados utilizado, foi configurado o parâmetro como **pgsql**;
- **Usuário, senha e servidor:** Indicam o usuário dono da base de dados, sua senha para acesso à base de dados e o local onde se encontra o servidor de banco de dados, respectivamente;
- **Base de dados:** Indica o nome da base de dados que software irá utilizar.

Com a base de dados criada e o arquivo **db.inc.php** configurado, o próximo passo foi criar a estrutura de tabelas necessárias ao funcionamento do *RoundCube*. Para isso foi utilizado o arquivo **postgres.initial.sql**, localizado em **/var/www/roundcubemail/SQL**. Esse arquivo possui comandos SQL para a criação de toda a estrutura de tabelas do software. Para a criação dessa estrutura, foi executado o seguinte comando:

```
# psql -U roundcube roundcubemail < /var/www/roundcubemail/SQL/postgres.initial.sql
```

O comando **psql** é utilizado para se conectar em uma base de dados do PostgreSQL. A opção (**-U roundcube**) indica que o usuário **roundcube** será utilizado para se conectar na base de dados. Os parâmetros restantes indicam que os comandos SQL contidos no arquivo **postgres.initial.sql** devem ser executados na base de dados **roundcubemail**.

Depois de feita a criação da estrutura de tabelas do *RoundCube*, foi editado o arquivo de configuração **main.inc.php**. As principais linhas configuradas foram:

```
/var/www/roundcubemail/config/main.in.php:  
$rcmail_config['default_host'] = 'localhost';  
$rcmail_config['default_port'] = 143;  
$rcmail_config['smtp_server'] = localhost;  
$rcmail_config['smtp_port'] = 25;
```

Os parâmetros **default_host** e **default_port** configuram o *webmail* para conexão no servidor IMAP. Através da configuração desses dois parâmetros é possível realizar o *login* no *webmail* e visualizar as mensagens. Já os parâmetros **smtp_server** e **smtp_port** configuram o *webmail* para conexão no servidor SMTP, utilizado para o envio de e-mails.

Após a configuração do RoundCube, foi necessária a criação de um novo VirtualHost no servidor apache para ser possível o acesso ao webmail via navegador web. Foi criado o arquivo **roundcubemail.conf** dentro de **/etc/httpd/conf/vhosts.d** – conforme foi realizado para a configuração do Postfix Admin, no tópico 4.2.3.1 – com o seguinte conteúdo:

/etc/httpd/conf/vhosts.d/roundcubemail.conf:

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerAdmin webmaster@roundcubemail
    DocumentRoot /var/www/roundcubemail
    ServerName webmail.projetofinal.org
    ErrorLog /var/log/httpd/roundcubemail-error_log
    CustomLog /var/log/httpd/roundcubemail-access_log common
</VirtualHost>
```

O conteúdo do arquivo é bastante semelhante ao do arquivo **postfixadmin.conf**, criado durante a configuração do software Postfix Admin, porém é importante notar os parâmetros **DocumentRoot** e **ServerName**. Neste caso o primeiro aponta para o diretório do *RoundCube*, **/var/www/roundcubemail**, e o segundo está configurado como **webmail.projetofinal.org**.

Sendo assim, para acessar o *webmail* é necessário acessar a url **webmail.projetofinal.org** em um navegador de internet. A figura 26 mostra a tela de *login* do *RoundCube*, em um navegador de internet.

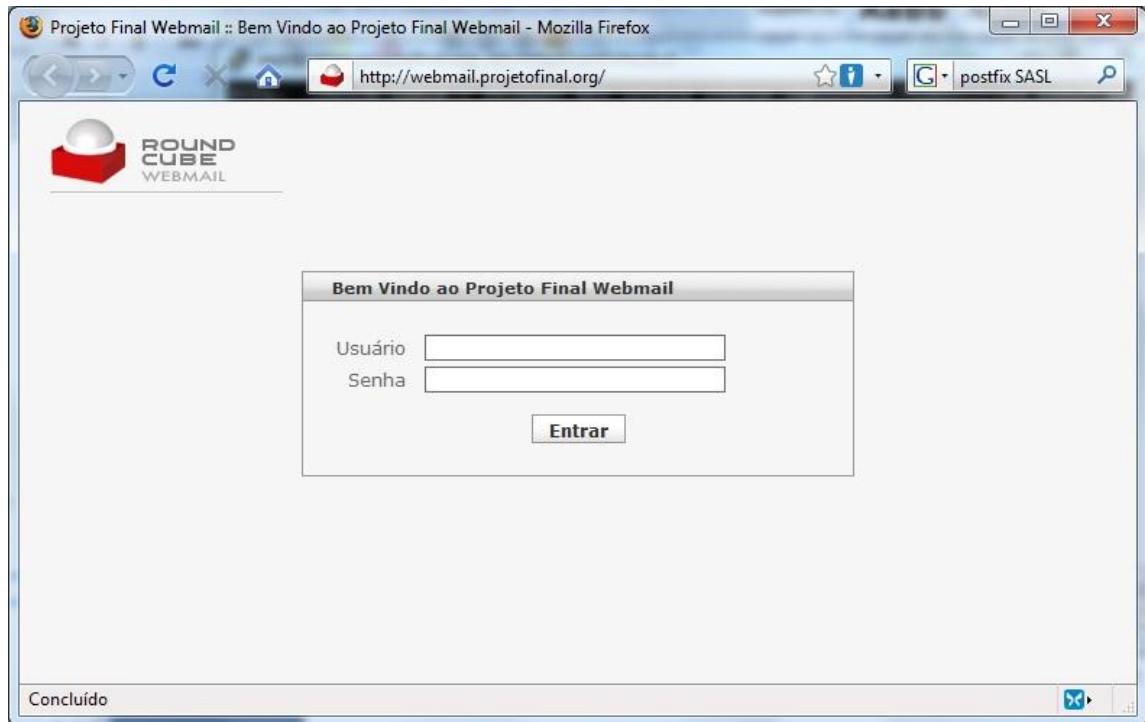


Figura 26: “Tela do login do webmail RoundCube.”

4.2.4. Implementação dos recursos de segurança

Neste tópico são abordadas as configurações que foram feitas para incrementar a segurança do servidor. São mostradas as configurações dos diversos filtros utilizados no combate aos spams, além das configurações de criptografia para manter o sigilo dos dados.

4.2.4.1. Restrições de recebimento de mensagens no Postfix e o SASL

O software Postfix possui configurações que podem atuar como uma primeira camada de filtro contra os spams. Através de verificações nos comandos SMTP recebidos, é possível eliminar diversas mensagens indesejadas logo no processo de recebimento da mensagem.

Para adicionar as restrições no software Postfix, foi adicionado o seguinte trecho no arquivo **main.cf**:

/etc/postfix/main.cf:

```
smtpd_client_restrictions =  
    reject_unknown_client  
smtpd_sender_restrictions =  
    reject_unknown_sender_domain  
smtpd_reject_unlisted_recipient = yes
```

O parâmetro **smtpd_client_restrictions** indica restrições que serão aplicadas no momento da conexão do remetente no servidor. Esse parâmetro vem seguido da restrição **reject_unknown_client**, que rejeita conexões de endereços IP que não possuem um registro de DNS reverso válido configurado. Sendo assim, no momento em que é feita uma conexão no servidor, o Postfix verifica se o endereço IP do remetente possui registro de DNS reverso válido, caso não possua, a mensagem é descartada.

Já o parâmetro **smtpd_sender_restrictions** indica restrições que são aplicadas após o recebimento do comando SMTP MAIL FROM. O mesmo é seguido da restrição **reject_unknown_sender_domain**, cuja função é mensagens enviadas de remetentes com um domínio que não é válido na internet.

Por fim, o parâmetro **smtpd_reject_unlisted_recipient**, rejeita mensagens que sejam enviadas para endereços de e-mail inexistentes na base de dados do servidor. A verificação desse parâmetro é realizada a partir dos dados recebidos no comando SMTP RCPT TO.

Após a configuração das restrições no Postfix, foi implementado no servidor o SASL (**Simple Authentication and Security Layer**). Através do SASL é criado um serviço de autenticação para o envio de mensagens a partir do servidor. Isso evita com que usuários ou códigos maliciosos utilizem o servidor para enviar spams pela internet.

O software Dovecot possui a funcionalidade de prover esse mecanismo de autenticação. Para isso foram adicionadas as seguintes linhas ao arquivo de configuração **dovecot.conf**, situado em **/etc**:

/etc/dovecot.conf:

```
client {
    path = /var/spool/postfix/private/auth
    user = postfix
    group = postfix
}
```

O trecho acima mostra a criação de um socket¹² de comunicação que será utilizado pelo Postfix para autenticar os usuários durante o envio de e-mails. Os parâmetros `user` e `group` indicam o usuário e grupo dono do socket, respectivamente. Já o parâmetro `path` indica o local onde será criado o socket.

Após a configuração do Dovecot, foi necessário fazer a integração desse novo recurso com o Postfix. Para isso, foi adicionado o seguinte trecho no arquivo **main.cf**:

/etc/postfix/main.cf:

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_type         = dovecot
smtpd_sasl_path         = /var/spool/postfix/private/auth
```

O parâmetro **smtpd_sasl_auth_enable** habilita o SASL no Postfix. Já o parâmetro **smtpd_sasl_type**, configurado com o valor `dovecot`, indica que o Postfix utilizará o dovecot para realizar a autenticação. Por fim, o parâmetro **smtpd_sasl_path** indica o local onde se encontra o socket de comunicação criado pelo Dovecot.

4.2.4.2. Configuração do SPF e do Postgrey

Os filtros SPF (*Sender Policy Framework*) e Postgrey são integrados ao software Postfix para auxiliar no combate aos spams.

¹² Refere-se ao socket Unix. Maiores informações em:

<<http://beej.us/guide/bgipc/output/html/multipage/unixsock.html>>. Acesso em 19 mai 2009.

Spam é um termo utilizado para se referir a mensagens que não foram solicitadas. Geralmente essas mensagens são anúncios publicitários. Um dos principais problemas gerados pelos spams é a degradação do desempenho das redes de comunicação e dos sistemas. [CGI.BR, 2009]

O filtro SPF baseia-se na verificação de MTAs válidos através dos registros no servidor DNS do domínio em questão. Para o correto funcionamento do SPF é necessário: [CGI.BR, 2009]

1. A configuração, no servidor DNS responsável pelo domínio, dos endereços IP referentes aos MTAs autorizados a enviar mensagens em nome do domínio em questão; [CGI.BR, 2009]
2. Configurações nos MTAs para que verifiquem junto aos registros DNS do domínio se o servidor que está tentando enviar o e-mail está autorizado. [CGI.BR, 2009]

Caso uma dessas configurações citadas acima não esteja correta ou não exista, o SPF terá o seu funcionamento comprometido.

A partir da figura 27 é possível verificar o registro TXT do domínio **uol.com.br**. Nesse registro encontra-se a configuração do SPF, indicando a faixa de IP dos MTAs autorizados pelo domínio **uol.com.br** para enviar e-mails em seu nome.

```
[root@server ~]# dig -t txt uol.com.br

; <<>> DiG 9.5.0-P2 <<>> -t txt uol.com.br
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49753
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;uol.com.br.                IN      TXT

;; ANSWER SECTION:
uol.com.br.                 3551    IN      TXT      "v=spf1 ip4:200.221.11.0/24
ip4:200.221.29.0/24 ip4:200.221.4.0/24 -all"
```

Figura 27: "Registro de MTAs autorizados pelo domínio uol.com.br."

No servidor do projeto foi configurado o cliente SPF. A função desse cliente é verificar se o MTA remetente é autorizado pelo domínio no qual ele está tentando enviar o e-mail. Essa verificação é feita através dos seguintes passos:

1. O MTA que está recebendo a mensagem verifica o domínio do remetente através do comando SMTP MAIL FROM recebido. Os comandos SMTP foram abordados no tópico 3.2.2.
2. Tendo posse do domínio do remetente, o MTA realiza uma verificação junto aos registros TXT do domínio se o endereço IP do MTA no qual está tentando enviar o e-mail está autorizado pelo domínio. Se o MTA estiver no registro, o e-mail é entregue; caso contrário, a mensagem é descartada.

Para a configuração do SPF no Postfix, foi adicionado o seguinte trecho no arquivo de configuração **main.cf**, situado em `/etc/postfix`:

/etc/postfix/main.cf:

```
smtpd_sender_restrictions =
    check_policy_service unix:private/spf,
```

O parâmetro **smtpd_sender_restrictions** indica as restrições do comando MAIL FROM, conforme já explicado tópico 4.2.4.1. Já o parâmetro **check_policy_service** configura uma política de segurança para o valor recebido em MAIL FROM. O valor indicado em **unix:private/spf** indica uma configuração que deverá ser buscada no arquivo de configuração **master.cf**, situado em `/etc/postfix`:

/etc/postfix/master.cf:

```
spf unix - n n - - spawn
    user=nobody argv=/usr/bin/perl /usr/lib/postfix/postfix-policyd-spf-perl
```

A configuração acima faz com que o Postfix execute o script Perl¹³ postfix-policyd-spf-perl. A função desse script é realizar a verificação do remetente conforme

¹³ Linguagem de programação criada por Larry Wall. Maiores informações em: <http://www.perl.org/about.html>. Acesso em 17 mai 2009.

já foi explicado anteriormente. O script Perl foi baixado diretamente no site do projeto do SPF, disponível em: <http://www.openspf.org/Software>.

Após a configuração do SPF, foi instalado o software Postgrey. Ele possui a função de implementar o conceito de Greylisting, que consiste em recusar uma mensagem temporariamente e aguardar pela sua retransmissão. [CGI.BR, 2009]

O Greylisting parte dos seguintes princípios:

1. Mensagens de e-mail válidas normalmente são enviadas de MTAs legítimos, que possuem mecanismos de retransmissão caso um erro temporário ocorra; [CGI.BR, 2009]
2. Spams raramente são enviados a partir de MTAs legítimos. [CGI.BR, 2009]

A principal vantagem do Greylisting é a redução sensível da quantidade de spams, uma vez que, como os spams geralmente não são enviados de MTAs legítimos, eles não são retransmitidos e, portanto, não chegam à caixa postal do usuário. Uma desvantagem é o fato de a mensagem levar um tempo maior que o normal para chegar à caixa postal do destinatário, visto que a mensagem é recusada na primeira vez que chega ao destino. A figura 28 mostra um diagrama de blocos com lógica geralmente implementada nos MTAs com suporte ao Greylisting:

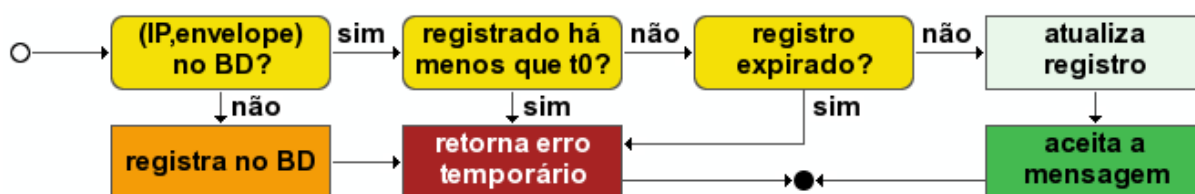


Figura 28: “Lógica implementada nos MTAs com suporte ao Greylisting.”

Fonte: <http://www.antispam.br/admin/greylisting/>

Ainda na figura 28, é possível verificar que, uma mensagem retransmitida em um tempo menor que **t0** é descartada e, caso uma mensagem seja retransmitida depois de um tempo muito longo, ou seja, o seu registro foi expirado, ela também

será descartada. Logo, existe um intervalo de tempo no qual a mensagem deve ser retransmitida para que ela seja entregue. Além da lógica abordada na figura 28, os MTAs com suporte ao Greylisting possuem as listas brancas. Os endereços contidos nessas listas são entregues automaticamente. As listas brancas são preenchidas da seguinte forma:

- **Automaticamente:** caso um mesmo remetente foi aprovado pelo mecanismo de Greylisting após sucessivos envios de mensagens;
- **Manualmente:** pelo administrador do sistema que informa os endereços de e-mail confiáveis na lista.

Para a instalação do software Postgrey, foi executado o seguinte comando:

```
# urpmi postgrey
```

O arquivo de configuração do Postgrey é o **postgrey**, situado em `/etc/sysconfig`. A configuração realizada nesse arquivo foi:

```
/etc/sysconfig/postgrey:
```

```
OPTIONS="--inet=localhost:10031 --auto-whitelist-clients"
```

O parâmetro **OPTIONS** indica quais opções o Postgrey deve habilitar. O valor **--inet=localhost:10031** configura o Postgrey para aguardar por requisições na porta TCP 10031. Já o valor **--auto-whitelist-clients** habilita as listas brancas.

Após a configuração do Postgrey foi necessário integrá-lo ao Postfix. Para isso foi adicionada a seguinte linha no arquivo **main.cf**:

```
/etc/postfix/main.cf:
```

```
smtpd_sender_restrictions =
```

```
    check_policy_service unix:private/spf,
```

```
    check_policy_service inet:localhost:10031,
```

O parâmetro **check_policy_service** foi adicionado novamente, logo abaixo do parâmetro que configura o SPF. A diferença pode ser observada no valor do parâmetro: **inet:localhost:10031**. Esse valor configura o Postfix para realizar uma requisição ao Postgrey, que irá validar o remetente da mensagem conforme já foi explicado.

É importante notar que, as políticas de segurança do SPF e do Postgrey, configuradas no Postfix, são sucessivas. Isso significa que uma mensagem, ao chegar no servidor de e-mail, é verificada na política do SPF e logo após na política do Postgrey. Esse dois filtros somados buscam reduzir de forma eficiente que mensagens indesejadas cheguem às caixas postais dos usuários.

4.2.4.3. Configuração do DSPAM

O último filtro a ser configurado no servidor é o DSPAM. Ele é um software que analisa o conteúdo das mensagens através de filtros bayesianos.

Os filtros Bayesianos utilizam um algoritmo de probabilidade com base na Teoria de Bayes. Geralmente, os softwares que utilizam esse tipo de filtro necessitam de um treinamento inicial, que analisa conjuntos de mensagens legítimas e indesejadas. A partir daí, o filtro verifica as próximas mensagens a partir da base de dados construída no treinamento inicial. Para que o filtro continue se aperfeiçoando, é necessário que o treinamento do mesmo seja continuado com as novas mensagens. [CGI.BR, 2009]

É importante notar que pode haver falsos positivos, ou seja, mensagens que foram classificadas como spams, porém são de interesse ao usuário. Por isso, não se deve descartar as mensagens automaticamente e sim, criar mecanismos de quarentena, para que o próprio usuário possa verificar se a mensagem é de fato legítima ou um spam. [CGI.BR, 2009]

Para a configuração do software DSPAM, foi necessária a criação de uma base de dados. A função principal da mesma é armazenar os dados estatísticos criados pelo DSPAM de acordo com o seu treinamento. Assim como nas outras

bases de dados criadas na implementação até então, foi criado um usuário com permissão total apenas à base de dados, chamada dspam. O nome denominado a esse usuário foi dspam. Depois de criado o usuário, foi criada a base de dados. A figura 29 mostra os comandos SQL executados para a criação do usuário dspam, além da base de dados dspam.

```
[root@server dspam]# psql -U postgres
Bem vindo ao psql 8.2.10, o terminal iterativo do PostgreSQL.

Digite: \copyright para mostrar termos de distribuição
        \h para ajuda com comandos SQL
        \? para ajuda com comandos do psql
        \g ou terminar com ponto-e-vírgula para executar a consulta
        \q para sair

postgres=# CREATE USER dspam WITH PASSWORD '3ed4rf5tg';
CREATE ROLE
postgres=# CREATE DATABASE dspam OWNER dspam;
CREATE DATABASE
postgres=#
```

Figura 29: “Criação da base de dados dspam.”

Depois de criada a base de dados, foi instalado o software DSPAM com o seguinte comando:

```
# urpmi dspam dspam-cgi
```

Depois de instalado o DSPAM, foi criado o usuário chamado **dspam** no sistema. Esse usuário tem como diretório pessoal **/var/lib/dspam**, onde são guardadas configurações e arquivos de log do DSPAM, além de dados estatísticos, como por exemplo, o percentual de acertos - mensagens que foram classificadas como spam corretamente - de um usuário. A função desse usuário é limitar o acesso do software DSPAM somente ao diretório **/var/lib/dspam**, por questões de segurança.

O próximo passo a ser executado foi a configuração do arquivo **dspam.conf**, situado em **/etc**. As principais linhas configuradas foram:

/etc/dspam.conf:

```

Home /var/lib/dspam
TrustedDeliveryAgent "/usr/lib/dovecot/deliver -d %u"
QuarantineAgent "/usr/lib/dovecot/deliver -d %u -m Spam"
PgSQLServer    localhost
PgSQLPort      5432
PgSQLUser      dspam
PgSQLPass      3ed4rf5tg
PgSQLDb        dspam

```

O parâmetro **Home** indica o diretório onde são guardados arquivos manipulados pelo DSPAM, tais como arquivos de log, configurações e dados estatísticos.

O parâmetro **TrustedDeliveryAgent** indica um agente de entrega confiável pelo DSPAM. Através desse parâmetro é configurado para quem o DSPAM entrega a mensagem caso a mesma tenha sido considerada legítima. O valor configurado para esse parâmetro ("/usr/lib/dovecot/deliver -d %u") indica que o DSPAM entrega a mensagem para o Dovecot, que por sua vez, armazena a mesma na caixa postal do usuário. Já o parâmetro **QuarantineAgent** configura para quem o DSPAM entrega a mensagem caso a mesma tenha sido classificada como spam. O valor configurado para esse parâmetro ("/usr/lib/dovecot/deliver -d %u -m Spam") é um pouco diferente quando comparado ao definido em **TrustedDeliveryAgent**. Nesse caso, a mensagem é entregue ao Dovecot, que por sua vez armazena a mesma em uma pasta especial, chamada Spam, na caixa postal do usuário. Com isso, os usuários podem verificar as mensagens que foram classificadas como spam em suas próprias caixas postais.

Os parâmetros **PgSQLServer**, **PgSQLPort**, **PgSQLUser**, **PgSQLPass** e **PgSQLDb** configuram o DSPAM para conectar-se à base de dados criada para sua utilização.

Após a configuração do arquivo dspam.conf, foi instalado o software de antivírus ClamAV com o seguinte comando:


```
# urpmi clamav
```

O ClamAV recebe requisições do DSPAM para realizar verificações de vírus nas mensagens. Seu arquivo de configuração é o **clamd.conf**, situado em **/etc**. As linhas configuradas nesse arquivo foram:

```
/etc/clamd.conf:  
TCPSocket 3310  
TCPAddr localhost  
ScanMail Yes
```

Os parâmetros **TCP Socket** e **TCPAddr** indicam, respectivamente, a porta e o endereço em que o ClamAV aguarda por requisições. Já o parâmetro **ScanMail** habilita o ClamAV para realizar a verificação em mensagens de e-mail.

Após a configuração do ClamAV, foi necessário integrá-lo ao DSPAM. Para isso foram adicionadas as seguintes linhas ao arquivo de configuração **dspam.conf**:

```
/etc/dspam.conf:  
ClamAVPort 3310  
ClamAVHost localhost
```

Essas linhas configuram os parâmetros necessários ao DSPAM para que o mesmo possa requisitar o ClamAV para realizar verificações de vírus nas mensagens.

Após isso foi integrado o DSPAM ao software Postfix. Para isso, foi adicionada a seguinte linha ao arquivo de configuração do Postfix **main.cf**:

```
/etc/postfix/main.cf:  
smtpd_client_restrictions =  
    reject_unknown_client,  
    check_client_access pcre:/etc/postfix/dspam_filter_access
```

O parâmetro **check_client_aces** neste caso faz com que o Postfix habilite o DSPAM para realizar a verificação da mensagem. É importante notar que esse

parâmetro foi adicionado logo abaixo do parâmetro **reject_unknow_client**, que já foi explicado no tópico 4.2.4.1, assim como o parâmetro **smtpd_client_restrictions**.

Depois disso, foi criado o arquivo **dspam_filter_access**, dentro do diretório **/etc/postfix**, com o seguinte conteúdo:

```
/etc/postfix/dspam_filter_access:
./ FILTER dspam:dspam
```

Após a criação desse arquivo, foi adicionado o seguinte trecho no arquivo **master.cf**, situado em **/etc/postfix**:

```
/etc/postfix/master.cf:
dspam unix      -      n      n      -      10      pipe
      flags=Ru user=dspam argv=/usr/bin/dspam
      --deliver=innocent,spam --user ${recipient}
```

O trecho acima indica o comando executado pelo Postfix para requisitar o DSPAM a verificar as mensagens.

O DSPAM possui um recurso importante que é o reaprendizado através do encaminhamento de mensagens. Caso uma mensagem seja classificada de forma incorreta, ela é encaminhada para um endereço específico e o DSPAM se encarrega de fazer a reaprendizagem, ou seja, classificar a mensagem de forma correta.

Para habilitar esse recurso, foi editado o arquivo de configuração **transport**, situado no diretório **/etc/postfix**:

```
/etc/postfix/transport:
spam@projetofinal.org      dspam-retrain:spam
ham@projetofinal.org      dspam-retrain:innocent
```

O trecho acima indica que as mensagens encaminhadas para o endereço **spam@projetofinal.org** são tratadas com o filtro **dspam-retrain:spam**. Esse filtro reaprende os falsos negativos, que são as mensagens indesejadas que não foram classificadas como spam. Já as mensagens encaminhadas para

ham@projetofinal.org são tratadas pelo filtro **dspam-retrain:innocent**. A função desse filtro é reaprender os falsos positivos, ou seja, as mensagens legítimas que foram classificadas como spam.

Após a edição do arquivo `transport`, foi necessário informar o comando a ser executado pelo Postfix para os filtros **dspam-retrain:spam** e **dspam-retrain:innocent**. Para isso, foi adicionado o seguinte trecho no arquivo **master.cf**:

/etc/postfix/master.cf:

```
dspam-retrain unix - n n - 2 pipe
flags=Ru user=dspam argv=/usr/bin/dspam --source=error
--class=${nexthop} --user ${sender}
```

O trecho acima indica que o Postfix irá executar o comando **/usr/bin/dspam** com os seguintes argumentos:

- **--source**: Indica o estado atual da classificação. O valor foi configurado como **error** para informar ao DSPAM que a classificação foi realizada de forma incorreta;
- **--class**: Indica qual é a classe da mensagem, ou seja, se ela é legítima ou spam. O valor foi configurado como **\${nexthop}**. Esse é substituído pelo valor configurado nos filtros **dspam-retrain**. Exemplo: caso o filtro seja **dspam-retrain:spam**, o valor **\${nexthop}** é substituído por **spam**;
- **--user**: Indica o usuário que está solicitando a reaprendizagem. **\${sender}** foi o valor configurado para esse parâmetro. Ele é substituído pelo endereço de e-mail do usuário no momento da execução do comando.

Outra forma de realizar a reaprendizagem das mensagens no software DSPAM é através de sua interface web. Os arquivos que compõem essa interface se encontram no diretório **/usr/share/dspam/cgi-bin**. Eles foram adicionados no momento da instalação do DSPAM.

Para utilizar essa interface web, foram necessárias configurações no servidor web Apache. Porém, não foi criado um Virtual Host específico para esse fim. As configurações foram realizadas no Virtual Host criado para utilização do Postfix Admin. A configuração realizada no Apache para o Postfix Admin foi abordada no tópico 4.2.3.1. O seguinte trecho foi adicionado no arquivo **postfixadmin.conf**, situado em **/etc/httpd/conf/vhosts.d**:

/etc/httpd/conf/vhosts.d/postfixadmin.conf:

```
Alias /dspam /usr/share/dspam/cgi-bin
<Directory /usr/share/dspam/cgi-bin/>
    Options ExecCGI
    DirectoryIndex dspam.cgi
    Auth_IMAP_Enabled on
    AuthName "Authorization required"
    AuthType Basic
    Auth_IMAP_Authoritative on
    Auth_IMAP_Server localhost
    Auth_IMAP_Port 143
</Directory>
```

O parâmetro **Alias** foi configurado de forma com que fosse criado um link, chamado **/dspam**, apontando para o diretório **/usr/share/dspam/cgi-bin**, local onde se encontram os arquivos da interface web do DSPAM. Como esse parâmetro foi configurado dentro do arquivo **postfixadmin.conf**, o caminho necessário para acessar a interface web será **http://admin.projetofinal.org/dspam**.

As configurações realizadas entre os parâmetros **<Directory /usr/share/dspam/cgi-bin/>** e **</Directory>** se restringem ao diretório **/usr/share/dspam/cgi-bin**. O quadro 5 descreve as funções desses parâmetros:

Quadro 5: “Parâmetros de autenticação para interface web do DSPAM.”

Parâmetro	Descrição
Options ExecCGI	Permite a execução de scripts.cgi
DirectoryIndex dspam.cgi	Indica o arquivo que deve ser executado ao acessar

	o diretório web.
Auth_IMAP_Enabled on	Habilita a autenticação IMAP.
AuthName "Authorization required"	Configura a frase que será exibida ao solicitar a autenticação.
AuthType Basic	Configura o tipo de autenticação como Basic.
Auth_IMAP_Authoritative on	Faz com que a autenticação IMAP tenha preferência sobre qualquer tipo de autenticação configurado posteriormente.
Auth_IMAP_Server localhost	Indica o servidor IMAP onde será feita a autenticação.
Auth_IMAP_Port 143	Indica a porta do servidor IMAP onde será feita a autenticação.

Como o tipo de autenticação para acesso foi configurado como IMAP, todos os usuários com contas de e-mail registradas no servidor podem acessar a interface web do DSPAM. Para testar esse acesso, foi criada a conta de e-mail **teste@exemplo.org** no servidor, através do software Postfix Admin. A figura 30 mostra a interface web do DSPAM, acessada através do usuário criado anteriormente:

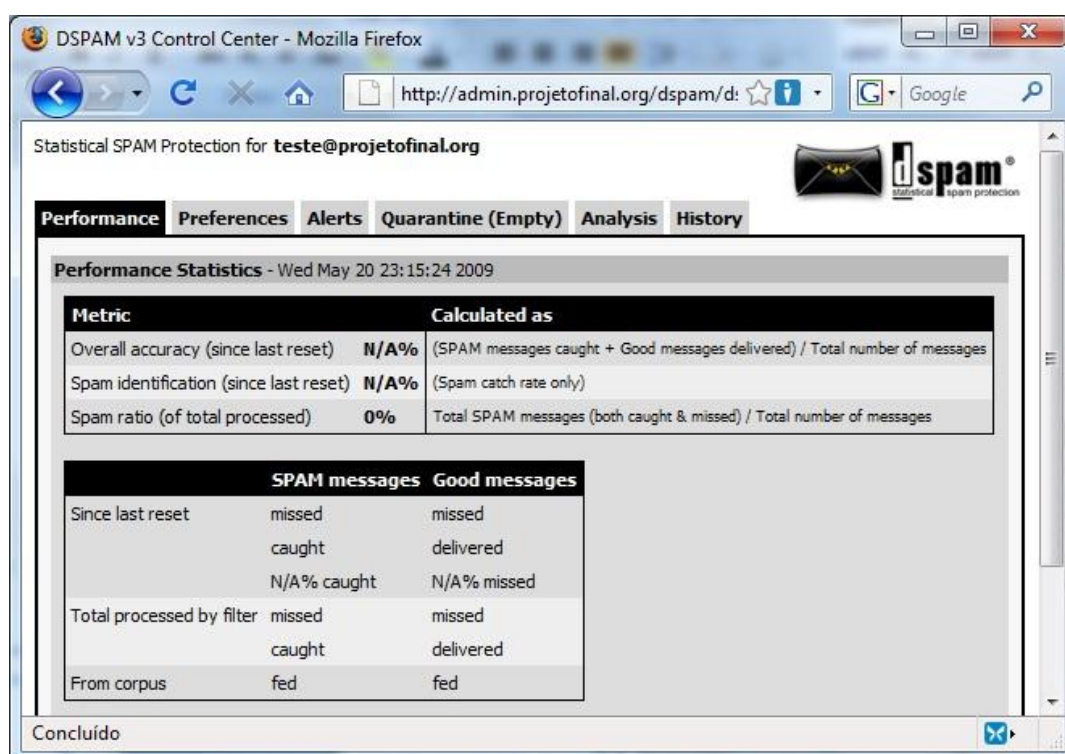


Figura 30: "Interface web do dspam."

4.2.4.4. Utilização dos protocolos HTTPS, POPS e IMAPS

É importante ressaltar que não é abordado detalhes sobre a criptografia utilizada para a geração destes certificados, visto que o propósito do projeto não é fazer uma análise sobre o melhor método de criptografia a ser configurado para esse tipo de solução.

Os protocolos HTTPS, POPS e IMAPS foram ativados utilizando certificados auto-assinados. Para a criação desses certificados, foi criado o arquivo **openssl.cnf**, dentro do diretório **/etc/ssl**. Esse arquivo contém informações sobre as características do certificado, como por exemplo, o CN (*Common Name*) do servidor para qual o certificado será emitido. As principais linhas configuradas foram:

```
/etc/ssl/openssl.cnf:  
C=BR  
ST=DF  
L=Brasilia  
O=Final Version  
OU=IT Department  
CN=mail.projetofinal.org
```

Os parâmetros C, ST e L indicam o país, o estado e a localidade de quem expediu o certificado, respectivamente. Já os parâmetros O, OU e CN indicam, respectivamente, a empresa, a unidade organizacional e o nome comum de quem expediu o certificado. Como os certificados são auto-assinados, os valores indicados em O, OU e CN valem também para quem o certificado foi expedido. É importante lembrar que o valor especificado em CN foi alterado para gerar três certificados diferentes:

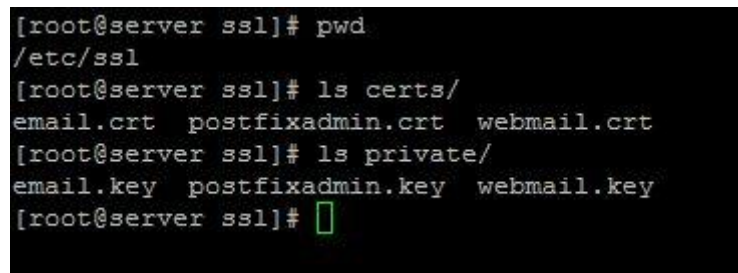
1. **mail.projetofinal.org**: certificado utilizado nos softwares Postfix e Dovecot para a ativação do TLS nos protocolos IMAP, POP e SMTP;
2. **webmail.projetofinal.org**: utilizado para utilização do protocolo HTTPS no acesso ao webmail;

3. **admin.projetofinal.org**: utilizado para utilização do protocolo HTTPS no acesso à ferramenta Postfix Admin.

Para a geração do certificado para **admin.projetofinal.org**, foi executado o seguinte comando:

```
# openssl req -new -x509 -config /etc/ssl/openssl.cnf -out /etc/ssl/certs/postfixadmin.crt -
keyout /etc/ssl/private/postfixadmin.key -days 3650
```

Através do comando acima, foi criado o certificado **postfixadmin.crt** no diretório **certs** e a chave privada **postfixadmin.key** foi guardada no diretório **private**, ambos situados em **/etc/ssl**. Esse comando foi repedido para a criação dos certificados para **webmail.projetofinal.org** e **mail.projetofinal.org**. A figura 31 mostra os certificados e as chaves privadas geradas, dentro de seus respectivos diretórios:



```
[root@server ssl]# pwd
/etc/ssl
[root@server ssl]# ls certs/
email.crt postfixadmin.crt webmail.crt
[root@server ssl]# ls private/
email.key postfixadmin.key webmail.key
[root@server ssl]#
```

Figura 31: “Certificados utilizados para os protocolos HTTPS, IMAPS e POPS”

Ainda na figura 31, nota-se a presença dos arquivos **email.crt** e **email.key**, criados para o CN **mail.projetofinal.org**, além dos arquivos **webmail.crt** e **webmail.key**, criados para o CN **webmail.projetofinal.org**.

Depois de gerados os certificados, foram configurados os softwares Apache, Postfix e Dovecot para utilização dos mesmos. Para o servidor web Apache, foi adicionado o seguinte trecho no arquivo de configuração **postfixadmin.conf**, que configura o *Virtual Host* **admin.projetofinal.org**:

/etc/httpd/conf/vhosts.d/postfixadmin.conf:

SSLEngine on

SSLCertificateFile /etc/ssl/projeto/certs/postfixadmin.crt

```
SSLCertificateKeyFile /etc/ssl/projeto/private/postfixadmin.key
```

O parâmetro **SSLEngine** possui a finalidade de habilitar o SSL na página. Já os parâmetros **SSLCertificateFile** e **SSLCertificateKeyFile** indicam, respectivamente, o certificado e a chave privada a serem utilizados. É importante notar que o certificado (**postfixadmin.crt**) e a chave privada (**postfixadmin.key**) selecionada configurada foram gerados para o CN **admin.projetofinal.org**. Esse valor coincide com o **ServerName** configurado nesse arquivo de configuração. O mesmo foi feito para o arquivo de configuração **roundcubemail.conf**, porém os arquivos especificados em **SSLCertificateFile** e **SSLCertificateKeyFile** apontam para **webmail.crt** e **webmail.key**, que foram gerados para o CN **webmail.projetofinal.org**.

Para a configuração do software Dovecot, foram adicionadas as seguintes linhas no arquivo de configuração **dovecot.conf**:

```
/etc/dovecot.conf:
```

```
ssl_disable = no
ssl_cert_file = /etc/ssl/projeto/certs/email.crt
ssl_key_file = /etc/ssl/projeto/private/email.key
```

O parâmetro **ssl_disable** indica se o Dovecot deve desabilitar o SSL. Já os parâmetros **ssl_cert_file** e **ssl_key_file** indicam o certificado e a chave privada a serem utilizados, respectivamente. Neste caso foi configurado o certificado e a chave gerada para o CN **mail.projetofinal.org**. Portanto, na configuração dos clientes de e-mail deve ser configurado como esse CN como o nome do servidor IMAP ou POP a ser utilizado.

Por fim, para a configuração do Postfix, foram adicionadas as seguintes linhas ao arquivo de configuração **main.cf**:

```
/etc/postfix/main.cf:
```

```
smtpd_use_tls = yes
```



```
smtpd_tls_cert_file = /etc/ssl/projeto/certs/email.crt  
smtpd_tls_key_file = /etc/ssl/projeto/private/email.key
```

O parâmetro **smtpd_use_tls** indica se o Postfix deve habilitar o TLS para o envio de e-mails. Já parâmetro **smtpd_tls_cert_file** indica o local do certificado a ser utilizado. Por fim, o parâmetro **smtpd_tls_key_file** indica o local da chave privada a ser utilizada. Novamente foi configurado o certificado e a chave gerada para o CN **mail.projetofinal.org**. Portanto, na configuração dos clientes de e-mail deve ser configurado como esse CN como o nome do servidor SMTP a ser utilizado.

4.3. Remasterização da imagem do sistema

Após a instalação de todos os recursos do servidor, foi iniciado o processo de remasterização da imagem de instalação do sistema. Para isso, foi criada uma estrutura de diretórios dentro do próprio servidor de e-mail. Ela foi criada no diretório chamado **/mnt/build**, diretório base criado para a remasterização da imagem original. Dentro desse diretório foram criados três subdiretórios:

- **iso**: Local onde se encontram as imagens ISO 9660 de instalação. Tanto a imagem original como a remasterizada foram arquivadas nesse diretório;
- **mandriva**: Local onde ficam os arquivos extraídos da imagem de instalação. É nesse local que o processo de remasterização ocorre;
- **temp**: Ponto de montagem temporário para extração dos arquivos da imagem de instalação. Entende-se como ponto de montagem um diretório onde é exibido o conteúdo de um dispositivo ou mídia, como por exemplo, um drive de CD-ROM ou uma imagem ISO 9660.

Depois de criada a estrutura de diretórios, foi inserida dentro do diretório **iso** a imagem de instalação original da distribuição Mandriva Linux 2009.0. Após isso, foi

feita a operação de montagem dessa imagem no diretório **temp**, para que seja exibido o conteúdo da mesma. Isso foi feito com o seguinte comando:

```
# mount -o loop iso/mandriva-linux-free-2009-dual-arch.iso temp/
```

O comando **mount** executado com a opção **-o loop** permite a montagem de imagens ISO 9660. O primeiro valor passado para o comando indica o caminho para a imagem a ser montada. Já o segundo valor indica o local onde a imagem é montada.

A partir daí, já é possível verificar todo o conteúdo da imagem de instalação do sistema no diretório **temp**. A figura 32 mostra o processo de montagem da imagem, além do seu conteúdo montado no diretório **temp**:

```
[root@server build]# mount
mount      mountpoint
[root@server build]# mount -o loop iso/mandriva-linux-free-2009-dual-arch-autoinstall.iso temp/
[root@server build]# cd temp/
[root@server temp]# pwd
/mnt/build/temp
[root@server temp]# ls
autorun.inf  dosutils/  i586/      lang/      TRANS.TBL  x86_64/
[root@server temp]#
```

Figura 32: “Processo de montagem da imagem de instalação.”

Ainda na figura 32, é possível perceber a existência dos diretórios **i586** e **x86_64**. Eles indicam os arquivos para a instalação em um computador com processador de 32 bits e 64 bits, respectivamente.

Após o processo de montagem, todo o conteúdo mostrado em **temp** foi copiado para o diretório **mandriva**, fazendo com que os arquivos da imagem fossem copiados para o disco rígido do servidor.

Com todo o conteúdo da imagem de instalação copiada, foi iniciado o processo de remasterização. Foi acessado o diretório **i586** para customizar a instalação do sistema para servidores com processador de 32 bits. A figura 33 mostra o conteúdo desse diretório:

```

[root@server i586]# pwd
/mnt/build/mandriva/i586
[root@server i586]# ls
COPYING          install/         media/           release-notes.html
default.xbe*     INSTALL.txt      misc/            release-notes.txt
doc/             isolinux/        pkg-2009-zarapha.idx  TRANS.TBL
export*          LICENSE.txt      product.id       VERSION
index.htm        linuxboot.cfg   README.txt
[root@server i586]#

```

Figura 33: “Diretório i586 situado na imagem de instalação do sistema.”

A partir da figura 33, verifica-se a existência dos diretórios **media**, **install** e **isolinux**. O primeiro contém toda a estrutura de diretórios que será criada durante a instalação do sistema. O segundo contém todos os pacotes RPM¹⁴ disponíveis para a instalação. Por fim, o terceiro possui arquivos para a inicialização do processo de instalação do sistema. Diante disso, o processo de customização foi dividido em três etapas:

1. Excluir todos os softwares desnecessários do diretório **media**;
2. Criar uma estrutura de diretórios, contendo todos os arquivos necessários para a configuração do servidor de e-mail, no diretório **install**;
3. Criar o script que automatiza o processo de instalação do sistema e configurar a sua execução no diretório **isolinux**.

Para selecionar os pacotes que seriam excluídos do diretório **media**, foi verificado no servidor todos os pacotes que foram instalados até então. Isso pode ser verificado a partir do seguinte comando:

```
# rpm -qa > /tmp/pacotes.txt
```

O comando **rpm** executado com opção **-qa** exibe todos os pacotes instalados no sistema. Já o símbolo de maior (>) possui a finalidade de redirecionar toda a

¹⁴ Formato de pacotes criado pela empresa Red Hat. Maiores informações em:
<http://docs.fedoraproject.org/drafts/rpm-guide-en/ch-intro-rpm.html>. Acesso em 23 mai 2009.

saída produzida pelo comando **rpm** para o arquivo chamado **pacotes.txt**, situado em **/tmp**. Tendo em posse a lista com todos os pacotes instalados, foi feita uma comparação com os pacotes existentes no diretório **media**. Os pacotes que não constavam em **pacotes.txt** foram excluídos.

Após isso, foi acessado o diretório **stage2**, situado em **install**. A figura 34 exibe a localização desse diretório, além de seu conteúdo:

```
[root@server stage2]# pwd
/mnt/build/mandriva/i586/install/stage2
[root@server stage2]# ls
mdkinst.sqfs*  rescue.sqfs*  TRANS.TBL  VERSION
[root@server stage2]#
```

Figura 34: “Conteúdo do diretório stage2.”

Ainda na figura 34, nota-se a existência do arquivo **mdkinst.sqfs**. Esse arquivo é uma imagem no formato squashfs¹⁵, com toda a estrutura de diretórios que é criada no servidor durante o processo de instalação. Para extrair o conteúdo dessa imagem, foi executado o seguinte comando:

```
# unsquashfs mdkinst.sqfs
```

Após a extração do conteúdo da imagem, foi criado um diretório chamado **squashfs-root**, contendo toda a estrutura de diretórios, já explicada anteriormente. A figura 35 exibe esse diretório, e todo o seu conteúdo:

```
[root@server stage2]# ls
mdkinst.sqfs*  rescue.sqfs*  squashfs-root/  TRANS.TBL  VERSION
[root@server stage2]# cd squashfs-root/
[root@server squashfs-root]# ls
bin/  etc/  lib/  opt/  sbin/  usr/
[root@server squashfs-root]#
```

Figura 35: “Estrutura do diretório squashfs-root.”

15 Maiores informações em: < <http://sourceforge.net/projects/squashfs/> >. Acesso em 23 mai 2009.

Ainda acerca da figura 35, verifica-se a existência do diretório **opt**. Dentro dele foi criado o diretório **configurador**, que contém os seguintes subdiretórios: **bin**, **confs**, **dmpps** e **rpms**.

O subdiretório **bin** contém os arquivos do script que customizam a instalação do sistema. Esse assunto é abordado no próximo tópico. Já o subdiretório **rpms** contém os pacotes no formato RPM dos softwares instalados durante a configuração manual do servidor, descrita no tópico 4.2.

O subdiretório **dumps** possui arquivos com comandos SQL para a criação da estrutura de tabelas das bases de dados **postfix**, **roundcubemail** e **dspam**, criadas para o funcionamento dos softwares instalados no servidor. No caso das bases de dados **dspam** e **roundcubemail**, foram copiados os arquivos com comandos SQL disponibilizados pelos próprios softwares, utilizados durante a configuração do servidor. Já no caso da base de dados **postfix**, foi feito um dump¹⁶ da mesma. Isso precisou ser feito devido ao fato de a ferramenta Postfix Admin criar a sua estrutura de tabelas através de um script PHP, executado via navegador de internet.

Por fim, o subdiretório **confs** possui todos os arquivos de configuração dos softwares utilizados no servidor. Foram copiados os arquivos que foram configurados durante a implementação do servidor. Sendo assim, durante o processo de automatização da configuração, será preciso realizar apenas alguns ajustes nesses arquivos.

A figura 36 mostra a estrutura de pastas criada dentro do diretório **configurador**:

¹⁶ O termo dump nesse caso trata-se de um backup no formato SQL Dump de uma base de dados no banco PostgreSQL. Maiores informações em: <<http://www.postgresql.org/docs/8.1/static/backup.html>>. Acesso em 23 mai 2009

```
[root@server configurador]# pwd
/mnt/build/mandriva/i586/install/stage2/squashfs-root/opt/configurador
[root@server configurador]# ls
bin/  confs/  dmps/  rpms/
[root@server configurador]# cd confs/
[root@server confs]# ls
apache/  dovecot/  dspam/  postfix/  postfixadmin/  roundcube/  ssl/
[root@server confs]#
```

Figura 36: “Estrutura de pastas do diretório configurador.”

Depois de criado o diretório configurador e toda a sua estrutura, foi necessário gerar a nova imagem squashfs. Para isso foi executado o seguinte comando:

```
# mksquashfs squashfs-root
```

O comando acima gera a imagem squashfs a partir do diretório base **squashfs-root**. Depois de gerada a imagem, foi removido o diretório squashfs-root para que o mesmo não fique ocupando espaço extra.

Depois de feita a exclusão dos softwares desnecessários e a criação da estrutura necessária para a configuração automatizada do sistema, o último passo foi a criação do script que automatiza a instalação do sistema. Uma característica importante da distribuição Mandriva Linux é que, após a instalação do sistema é criado o diretório /root/drakx, contendo arquivos que dizem a respeito da instalação realizada. Dentre esses arquivos, existe um chamado **auto_inst.cfg.pl**. Ele é um script em Perl que possui todas as configurações realizadas durante o processo de instalação do sistema. Através desse arquivo é possível automatizar a instalação de do sistema.

Diante disso, o arquivo **auto_inst.cfg.pl** foi renomeado para **auto_inst.cfg** e colocado dentro do diretório i586. Depois disso foi adicionado o seguinte trecho no arquivo **isolinux.cfg**, situado no diretório **isolinux**:

```
/mnt/build/mandriva/i586/isolinux/isolinux.cfg:
default autoinstall
label autoinstall
    kernel alt0/vmlinuz
```

```
append initrd=alt0/all.rdz automatic=method:cdrom vga=788 splash=silent dir=/i586
auto_install=auto_inst.cfg
```

O parâmetro **default** indica o tipo de inicialização padrão. Ele foi definido como **autoinstall** para que seja executada a inicialização especificada no parâmetro **label autoinstall**. Esse por sua vez, configura os parâmetros necessários para a inicialização do programa instalação. O parâmetro **kernel** informa o local onde se encontra o kernel que é carregado na inicialização. Por fim, o parâmetro **append** indica as configurações para a inicialização do programa de instalação. É importante perceber a existência dos valores **dir=/i586** e **auto_install=auto_inst.cfg**, pois são eles que informam ao programa de instalação que ele deverá realizar uma instalação automática, baseadas nas configurações existentes no arquivo **auto_inst.cfg**.

Depois de executados todos os passos para a remasterização no diretório i586, referente à instalação de um sistema 32 bits, o mesmo foi realizado no diretório x86_64, que diz respeito à instalação de um sistema 64 bits.

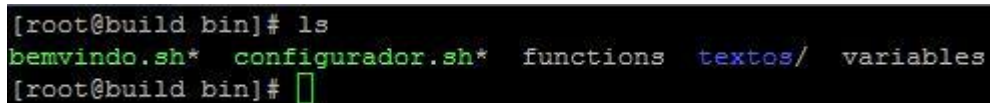
Após isso, foi gerada a imagem ISO 9660 remasterizada, a partir do seguinte comando:

```
# mkisofs -o /mnt/build/iso/mandriva-linux-2009-dual-arch-autoinst.iso /mnt/build/mandriva
```

O comando **mkisofs** gera uma imagem ISO 9660 com o nome **mandriva-linux-2009-dual-arch-autoinst.iso** no diretório **/mnt/build/iso** a partir do conteúdo do diretório **/mnt/build/mandriva**.

4.4. Automatização da configuração do servidor

Antes de ser finalizada a imagem remasterizada, pronta para ser utilizada em uma instalação, foi criado o script que automatiza o processo de configuração do servidor. Os arquivos desse script foram inseridos dentro do diretório **bin**, situado no diretório configurador, conforme já explicado no tópico anterior. A partir da figura 37, é possível notar a existência dos arquivos que compõem o script de automatização da configuração do servidor, situados no diretório **bin**:



```
[root@build bin]# ls
bemvindo.sh*  configurador.sh*  functions  textos/  variables
[root@build bin]#
```

Figura 37: “Conteúdo do diretório bin.”

Ainda na figura 37, verifica-se a existência dos arquivos **bemvindo.sh** e **configurador.sh**. O primeiro é o script que será executado na primeira inicialização do sistema. Sua função é passar as informações necessárias a quem está realizando a instalação além de realizar as configurações iniciais do sistema, como por exemplo, as configurações de rede. Já o segundo é o script principal, que realiza as configurações necessárias ao funcionamento do servidor. Existem também os arquivos **functions** e **variables**, que indicam as funções chamadas pelo script principal e as variáveis necessárias para a execução das funções, respectivamente. Por fim, existe o diretório **textos**, cuja função é armazenar arquivos os textos explicativos utilizados no script **bemvindo.sh**.

O primeiro arquivo a ser abordado é o **variables**. Sua função é armazenar as variáveis necessárias para a execução das funções criadas dentro do arquivo **functions**. Algumas variáveis contidas nesse arquivo devem ser editadas pelo administrador que está realizando a instalação do sistema. Diante desse fato, todo o arquivo foi comentado, indicando quais variáveis deverão ser definidas pelo administrador, além das funções delas. As linhas desse arquivo que devem ser editadas são:

/opt/configurador/bin/variables:

DOM_NAME=projetofinal.org

FILTERS=N

CERT_C=BR

CERT_ST=DF

CERT_L="Brasilia"

CERT_O="Projeto Final"

CERT_OU="Projeto Final"

O trecho acima mostra as variáveis já com valores definidos, porém eles devem ser alterados para atender as necessidades do administrador que está configurando o servidor. A variável **DOM_NAME** indica qual é o domínio principal a ser configurado no servidor. A variável **FILTERS** indica se os filtros SPF e Postgrey devem ser utilizados ou não. Os valores possíveis para essa variável são: **N**, para desabilitar os filtros; e **Y** para desabilitar os filtros. As variáveis iniciadas por **CERT_** configuram os parâmetros necessários para a geração dos certificados auto-assinados. As variáveis **CERT_C**, **CERT_ST** e **CERT_L** indicam, respectivamente: a o país, o estado e a cidade da empresa. Já as variáveis **CERT_O** e **CERT_OU** indicam o nome da empresa e o nome da unidade organizacional, respectivamente.

Além das variáveis que devem ser editadas, existem outras que são importantes para o funcionamento correto das funções configuradas no arquivo **functions**. Elas definem os caminhos completos para os subdiretórios criados dentro do diretório **configurador**. São elas:

/opt/configurador/bin/variables:

CONF_DIR=/opt/configurador/confs

DMP_DIR=/opt/configurador/dmps

RPM_DIR=/opt/configurador/rpms

A variável **CONF_DIR** indica o caminho completo para o subdiretório onde se encontram os arquivos de configuração dos serviços utilizados pelo sistema. Já a variável **DMP_DIR** indica o local onde se encontra o subdiretório onde se encontram

os arquivos com comandos SQL para a construção das tabelas nas bases de dados utilizadas pelo servidor. Por fim, a variável **RPM_DIR** indica o local onde se encontram os pacotes no formato RPM dos softwares necessários ao funcionamento do servidor.

O arquivo **functions** possui todas as funções necessárias para a configuração do servidor. Essas funções são chamadas pelo script **configurador.sh**, que será abordado mais adiante. O trecho abaixo exhibe a função **configura_dovecot**, cujo objetivo é configurar o software Dovecot:

/opt/configurador/bin/functions:

```
1 configura_dovecot () {
2     echo "Configurando Dovecot.."
3     echo "-----"
4     sleep 1
5     getent passwd dspam || useradd -r -d /var/lib/dspam dspam
6     [ -f /etc/dovecot.conf ] && mv /etc/dovecot.conf /etc/dovecot.conf.old
7     [ -f /etc/dovecot-pgsql.conf ] && mv /etc/dovecot-pgsql.conf /etc/dovecot-pgsql.conf.old
8     cp $CONF_DIR/dovecot/* /etc
9     service dovecot restart
10 }
```

O comando **sleep**, executado na linha 4, tem a finalidade de parar a execução do script em 1 segundo. Isso foi feito para que possa ser observada a mensagem especificada nas linhas 2 e 3. Na linha 5 é executado o comando **getent passwd dspam** para verificar se o usuário **dspam** existe no sistema. O símbolo **||** indica que o próximo comando - que realiza a criação do usuário **dspam** - é executado somente se o primeiro retornar erro, que no caso seria a ausência do usuário **dspam** no sistema. Na linha 6 é verificado, através do comando **[-f /etc/dovecot.conf]**, se o arquivo **dovecot.conf**, situado em **/etc**, existe. O símbolo **&&** indica que o próximo comando - que renomeia o arquivo **dovecot.conf** para **dovecot.conf.old** - é executado somente se o primeiro não retornar erro, que no caso indica que o arquivo **dovecot.conf** existe de fato. A linha 7 possui a mesma função que a

anterior, exceto que a verificação no caso é do arquivo **dovecot-pgsql.conf**, situado em **/etc**. Na linha 8 são copiados todos os arquivos de configuração situados em **\$CONF_DIR/dovecot** para o diretório **/etc**. O valor **\$CONF_DIR** é substituído pelo valor configurado na variável **CONF_DIR**, que aponta para **/opt/configurador/confs**. Por fim, a linha 9 reinicia o Dovecot para que as novas configurações sejam carregadas.

O quadro 6 mostra todas as funções do arquivo **functions** e suas respectivas descrições.

Quadro 6: “Funções presentes no arquivo **functions**.”

Função	Descrição
edita_opensslcnf	Função auxiliar utilizada pela função configura_certificados . Possui a finalidade de editar o arquivo /etc/ssl/openssl.cnf .
cria_maildir_postmaster	Função auxiliar utilizada pela função configura_postfixadmin . Tem o objetivo de criar a caixa postal do usuário postmaster de acordo com o domínio especificado no arquivo variables .
cria_base_postfix	Função auxiliar utilizada pela função configura_postfix . Sua finalidade é criar a base de dados postfix .
cria_base_roundcubemail	Função auxiliar utilizada pela função configura_webmail . Sua finalidade é criar a base de dados roundcubemail .
cria_base_dspam	Função auxiliar utilizada pela função configura_antispam . Sua finalidade é criar a base de dados dspam .
configura_certificados	Cria os certificados auto-assinados.
configura_dovecot	Configura o software Dovecot.
configura_postfix	Configura o software Postfix.
configura_postfixadmin	Configura a ferramenta Postfix Admin.
configura_webmail	Configura o software de Webmail.
configura_filtros	Configura os filtros SPF e Postgrey
configura_antispam	Configura o software antispam DSPAM.

O próximo arquivo a ser abordado é o **configurador.sh**. Esse é o script principal, que utiliza as funções do arquivo **functions** para realizar toda a configuração do servidor. O trecho abaixo exhibe uma parte do script **configurador.sh**, que tem a finalidade de realizar verificações iniciais:

```
/opt/configurador/bin/configurador.sh:
1 if [ $USER != 'root' ];then
2     echo "Execute esse script como root"
3     exit 1
4 fi
5 if [ $# -ne 1 ];then
6     echo "Modo de uso: configurador.sh (certificados|filtros|full)"
7     exit 1
8 fi
9 source /opt/configurador/bin/functions
```

Nas linhas de 1 a 4 é verificado, através da estrutura **IF**, se o usuário que está executando o script - especificado na variável **USER** - é o **root**. Caso não seja, é exibida uma mensagem através do comando **echo**, informando que o script deve ser executado com o usuário **root**. Além disso, o script é finalizado com o valor 1, indicando erro. Já nas linhas de 5 a 8 é verificado se foi passado nenhum ou mais de um parâmetro após o comando que executa o script. Essa verificação é realizada através da expressão **\$# -ne 1**, onde **\$#** indica a quantidade de parâmetros passados e **-ne** significa “diferente de”. Caso nenhum ou mais um parâmetro tenha sido informado após o comando que executa o script, é exibida uma mensagem explicando a forma correta para se executar o script. Além disso, o script é finalizado com o valor 1. Por fim, na linha 9 são carregadas todas as funções presentes no arquivo **functions** através do comando **source**.

Após o trecho exibido acima, o script possui uma estrutura do tipo CASE, que executa as funções de acordo com o parâmetro informado após o comando que executa o script. Essa estrutura pode ser verificada no trecho abaixo:

/opt/configurador/bin/configurador.sh:

```

10 case $1 in
11     certificados)
12         configura_certificados
13         ;;
14     filtros)
15         configura_filtros
16         ;;
17     full)
18         configura_certificados
19         configura_dovecot
20         configura_postfix
21         configura_postfixadmin
22         configura_webmail
23         configura_filtros
24         configura_antispam
25         ;;
26     *)
27         echo "Modo de uso: configurador.sh (certificados|filtros|full)"
28         exit 1
29         ;;
30 esac

```

O parâmetro **\$1**, indicado na linha 1, indica qual foi o parâmetro passado. A partir daí, a estrutura CASE se encarrega de executar as funções adequadas de acordo com os parâmetros válidos informados: **certificados**, **filtros** ou **full**. O primeiro tem a finalidade de gerar os certificados auto-assinados apenas. Já o segundo tem o objetivo de configurar os filtros SPF e Postgrey. Por fim, o terceiro configura todos os serviços do servidor. Caso nenhum desses parâmetros seja informado é exibida uma mensagem explicando a forma correta para se executar o script, além de o script ser finalizado com o valor 1.

O último arquivo a ser abordado é o **bemvindo.sh**. Sua função é realizar as configurações iniciais do servidor, preparando-o para a execução do script **configurador.sh**, que configura o restante do servidor.

O script **bemvindo.sh** utiliza a ferramenta dialog para a criação de telas que possuem a finalidade de interagir de forma simples com o usuário. O seguinte trecho mostra as variáveis criadas no script:

```
/opt/configurador/bin/bemvindo.sh:
BASEDIR=/opt/configurador
BACKTITLE="Projeto Final - Script de configuração inicial - Thiago de Almeida
Milhomem"
SCREEN1=$(cat $BASEDIR/bin/textos/tela_inicial)
```

A variável **BASEDIR** indica o diretório base. Ela foi criada para evitar redigir o mesmo diretório várias vezes durante a criação do script. Já a variável **BACKTITLE** configura o texto que é exibido ao fundo das telas geradas pelo comando dialog. A variável **SCREEN1** recebe o conteúdo do arquivo **tela_inicial**, criado no diretório **/opt/configurador/bin/textos**. Esse arquivo contém um texto introdutório, que possui a finalidade de explicar a importância do script. Existem outras variáveis com finalidade semelhante a **SCREEN1**. A descrição de cada uma pode ser verificada no quadro 7:

Quadro 7: “Variáveis do script bemvindo.sh.”

Variável	Descrição
SCREEN2	Recebe o conteúdo do arquivo instalacao_pacotes .
SCREEN3	Recebe o conteúdo do arquivo alteracao_senha_root .
SCREEN4	Recebe o conteúdo do arquivo alteracao_senha_projeto .
SCREEN5	Recebe o conteúdo do arquivo configuracao_rede .
SCREEN6	Recebe o conteúdo do arquivo tela_final .

Todos os arquivos descritos no quadro 7 se encontram no diretório **/opt/configurador/bin/textos**, assim como o arquivo **tela_inicial**. O conteúdo completo desses arquivos pode ser observado no apêndice da monografia.

Depois de configurado as variáveis necessárias para o funcionamento do script, foram configuradas funções que possuem a finalidade de exibir as telas de

interação com o usuário através do comando **dialog**. O trecho abaixo exhibe a função **tela_inicial**:

```
/opt/configurador/bin/bemvindo.sh:
tela_inicial(){
    dialog --backtitle "$BACKTITLE" --title "Bem-vindo" --yes-label "OK" --no-label "Sair" --
yesno "$SCREEN1" 18 60
}
```

A função **tela_inicial** executa o comando **dialog** para gerar uma tela de interação com o usuário. O parâmetro **--backtitle** configura o título de fundo da tela gerada. Já o parâmetro **--title** configura o título da janela gerada. Os parâmetros **--yes-label** e **--no-label** indicam, respectivamente, o nome dos botões para os valores “sim” e “não” criados na janela gerada. O parâmetro **--yesno** indica o tipo de tela gerada pelo **dialog**. Esse tipo permite a escolha de duas opções: “sim” ou “não”. Através da escolha selecionada, o comando **dialog** retorna o valor 0, caso seja escolhida a opção “sim”; ou retorna o valor 1, caso a opção escolhida seja “não”. Por fim, os valores numéricos indicados ao final do comando configuram a largura e a altura da janela criada, respectivamente. A figura 38 mostra a tela inicial do script **bemvindo.sh**, gerada pela execução do comando **dialog** situado na função **tela_inicial**:

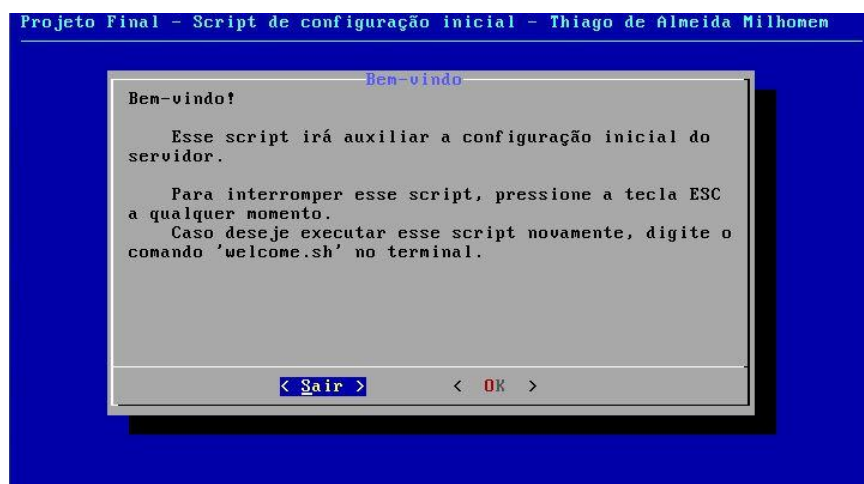


Figura 38: “Tela inicial do script bemvindo.sh.”

O quadro 8 mostra as outras funções do script **bemvindo.sh** e suas respectivas funções:

Quadro 8: “Funções do script *bemvindo.sh*.”

Função	Descrição
verifica_retorno	Verifica o valor retornado pelo comando <code>dialog</code> . Função necessária para realizar o processo de navegação entre as telas do script (ir e voltar).
instalacao_pacotes	Exibe uma tela informando que deverá realizar a instalação dos softwares necessários ao funcionamento do servidor.
mensagem_pacotes_instalados	Exibe uma mensagem informando que os softwares necessários ao funcionamento do servidor já estão instalados.
alteracao_senha_root	Mostra uma tela informando que deve ser configurada a senha do usuário <code>root</code> . Informa também que esse usuário possui privilégios administrativos no servidor.
alteracao_senha_projeto	Mostra uma tela informando que deve ser configurada a senha do usuário <code>projeto</code> . Informa que esse usuário pode ser utilizado para realizar acesso remoto no servidor, visto que isso não é possível pelo usuário <code>root</code> .
tela_final	Exibe uma tela informando que as configurações iniciais foram terminadas. Mostra novas instruções para continuar a configuração do servidor.

Depois de configuradas todas as funções, foi criado um menu sequencial, com a opção de ir e voltar, apresentando todas as telas configuradas nas funções do script, além de executar comandos que realizam a configuração inicial do servidor. Isso foi possível através da análise do valor de retorno do comando `dialog`. Caso o valor retornado seja 0, o menu avança para a próxima tela; caso o valor retornado seja 1 ou 255, o menu retrocede uma tela. O menu criado avança da seguinte maneira:

1. É exibida uma tela introdutória, informando que o script possui a finalidade de realizar configurações iniciais no servidor;
2. É mostrada uma tela informando que é necessária a instalação de softwares para o funcionamento correto do servidor. Após isso é verificado se os softwares já estão instalados, caso não, todos os pacotes situados em **/opt/configurador/rpms** são instalados;

3. É mostrada uma tela informando que é preciso configurar a senha do usuário **root**. Após isso é executado o comando **passwd root**, responsável por alterar a senha do mesmo;
4. É exibida uma tela informando que é preciso configurar a senha do usuário **projeto**. Depois disso é executado o comando **passwd projeto**, responsável por alterar a senha do mesmo;
5. É exibida uma tela informando que é necessário realizar as configurações de rede do servidor. Após isso, é executada a ferramenta **drakconnect**. Essa ferramenta foi criada pelo Mandriva e possui a finalidade de realizar diversos tipos de configuração de rede no sistema;
6. É mostrada a tela final do script, informando que a configuração ainda não foi finalizada. Informa também que para finalizar a configuração do servidor, é preciso verificar as informações existentes no arquivo **Leia-me.txt**, criado no diretório **/opt/configurador**.

Depois de configurado todos os arquivos para a automatização da configuração do servidor, foi criado o arquivo **Leia-me.txt** no diretório **/opt/configurador**. Esse arquivo possui a finalidade de passar informações acerca das configurações restantes que são necessárias para o funcionamento do servidor.

5. RESULTADOS OBTIDOS

5.1. Funcionamento do servidor

Para a realização dos testes com a solução implementada, o servidor foi configurado na internet. A figura 39 mostra a topologia utilizada para os testes:

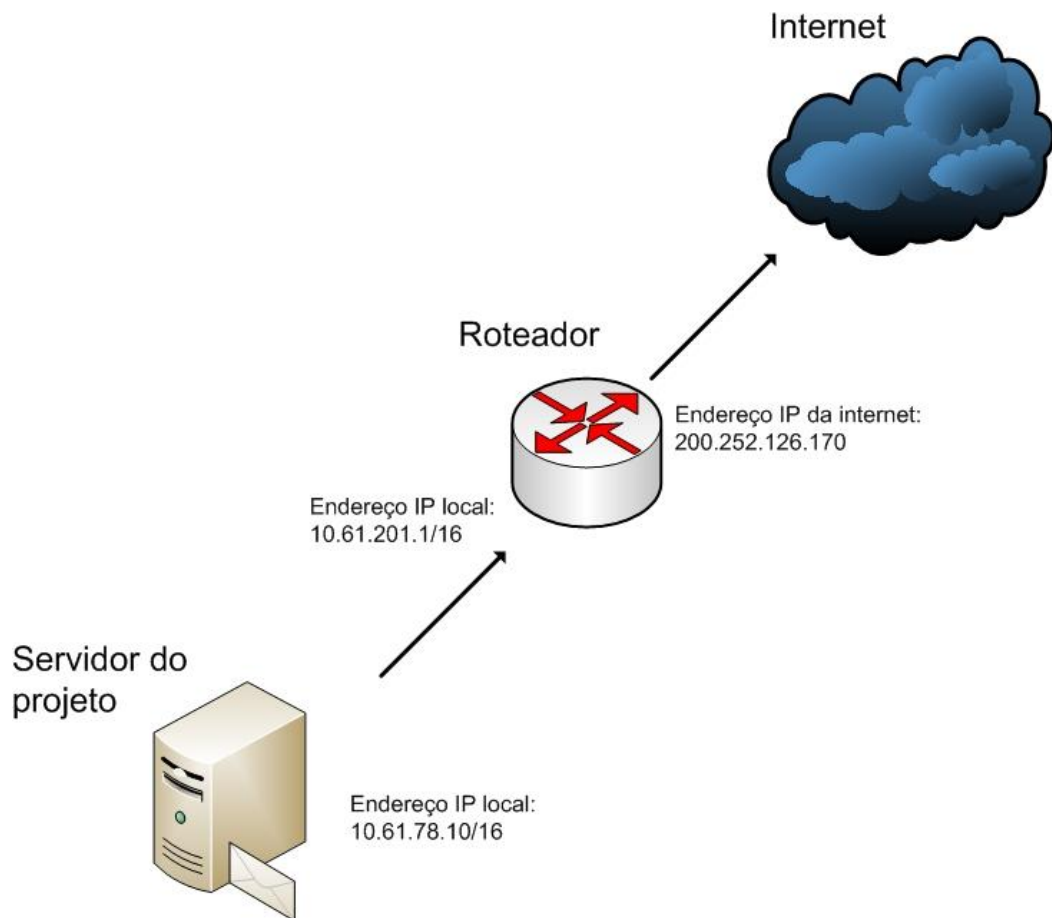


Figura 39: "Topologia utilizada nos testes."

Ainda na figura 39 é possível verificar que o endereço de rede configurado no servidor foi **10.61.78.10/16**. Além disso, percebe-se a presença de um roteador com o endereço IP local **10.61.201.1/16** e da internet **200.252.126.170**. Nas configurações do roteador foi direcionado todo o tráfego de entrada e saída do IP da internet para o servidor do projeto.

Depois de configurado o servidor na internet, foi cadastrado o domínio **projetofinal.org**. Nas configurações desse domínio, foram criados os seguintes registros DNS:

- Registro A **server.projetofinal.org**, apontando para o endereço IP da internet: **200.252.126.170**;
- Registros CNAME **admin.projetofinal.org**, **mail.projetofinal.org** e **webmail.projetofinal.org**. Todos eles foram configurados como apelidos para **server.projetofinal.org**;
- Registro MX do domínio configurado para **mail.projetofinal.org**;
- Registro PTR **200.252.126.170** apontando para **mail.projetofinal.org**.

Com essas configurações, o servidor se tornou apto para receber e enviar mensagens de outros domínios da internet.

Depois de realizadas as configurações de rede, foi acessada a ferramenta Postfix Admin através do endereço: **http://admin.projetofinal.org**. Foram criados os domínios **projetofinal.org** e **exemplo.com.br**. O segundo criado será utilizado como domínio secundário. As mensagens de teste enviadas para esse domínio precisam ser locais - enviadas do próprio servidor do projeto - visto que ele não foi registrado na internet. Após isso foram criadas as contas **teste@projetofinal.org** e **user@exemplo.com.br**. Elas foram criadas com a finalidade de testar o envio e recebimento de mensagens entre os domínios **exemplo.com.br** e **projetofinal.org**, ambos configurados no servidor. A conta **teste@projetofinal.org** foi utilizada também para verificar o envio e recebimento de e-mails para domínios da internet. A figura 40 mostra a interface web do Postfix Admin, além dos domínios criados:


postfix.admin

Administradores	Domínios	Virtual	Recuperar Mensagens	Enviar Mensagem	Alterar Senha	Histórico	Sair
-----------------	----------	---------	---------------------	-----------------	---------------	-----------	------

Domínio	Descrição	Aliases	Contas de Email	Cota de Espaço (MB)	MX de Backup	Última Modificação	Habilitado
exemplo.com.br	Domínio secundário	0 / 10	1 / 10	Ilimitado	Não	Wed Jun 3 22:17:13 2009 GMT	Sim Editar Remover
projetofinal.org	Domínio principal	0 / 0	1 / 0	Ilimitado	Não	Mon Apr 27 15:13:24 2009 GMT	Sim Editar Remover

[Criar Domínio](#)

Postfix Admin 2.3 rc2 | Autenticado como admin@projetofinal.org | Checar por atualização | [Return to admin.projetofinal.org](#)

Figura 40: “Domínios criados para testes na ferramenta Postfix Admin.”

Após isso, foi configurada a conta **user@exemplo.com.br** em um cliente de e-mail. Através dessa conta foi enviada uma mensagem para a conta **teste@projetofinal.org**. A partir da figura 41 é possível verificar o cliente de e-mail momentos antes de a mensagem ser enviada:

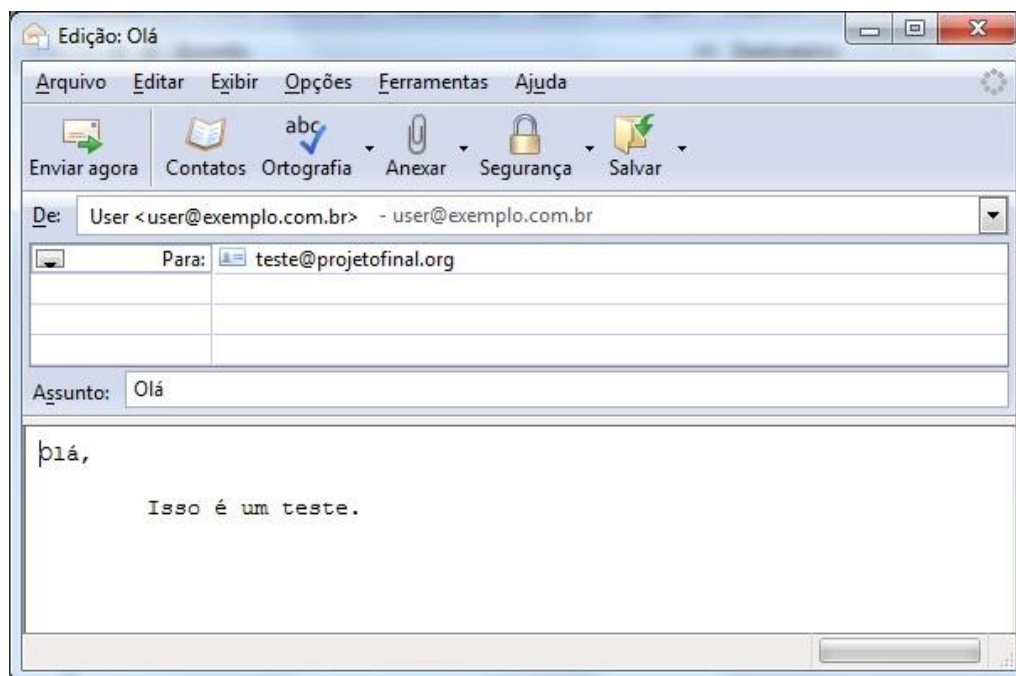


Figura 41: “Envio de uma mensagem de teste através de um cliente de e-mail.”

Depois de enviada a mensagem, foi acessada a conta **teste@projetofinal.org** através do webmail configurado no servidor, através do endereço **http://webmail.projetofinal.org**. A mensagem enviada pela conta **user@exemplo.com.br** foi recebida e visualizada no webmail, conforme pode ser observado na figura 42:

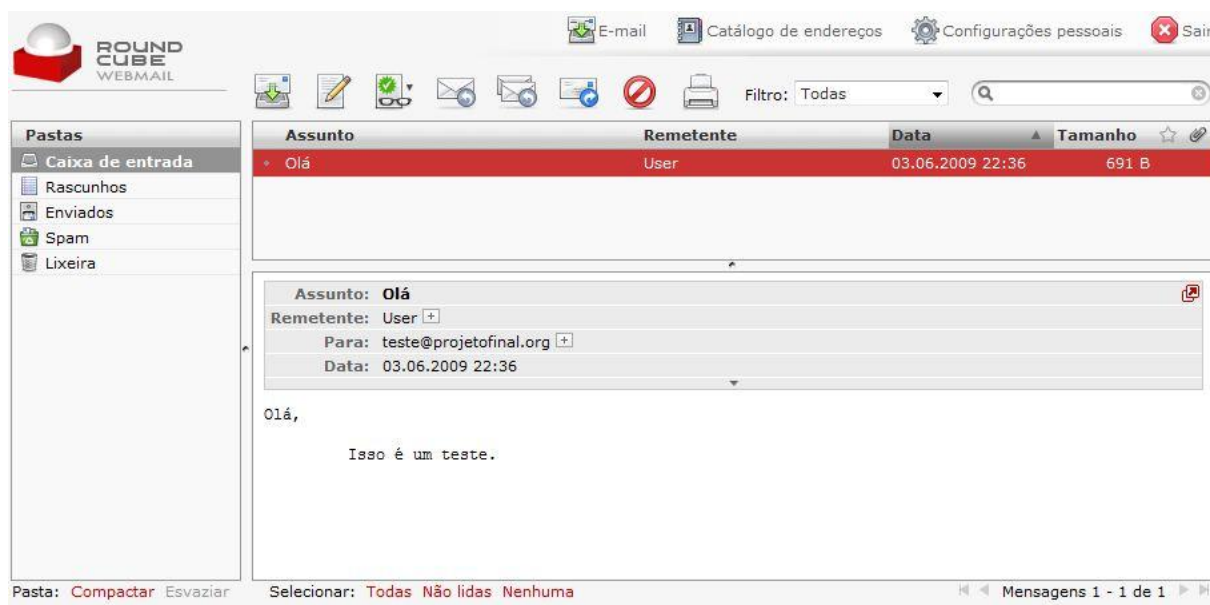


Figura 42: “Mensagem visualizada no webmail do projeto.”

Através desses testes realizados, foi possível verificar que o envio e recebimento local de mensagens, entre domínio de controle do servidor de e-mail, estavam funcionando. Foi possível verificar também que tanto a ferramenta Postfix Admin como o webmail foram configuradas corretamente.

Para o teste de envio para um domínio na internet, foi enviada uma mensagem através da conta **teste@projetofinal.org** para o e-mail **thiago@milhomem.org**. Essa conta de e-mail é administrada pela empresa Google, através do serviço Google Apps¹⁷. A figura 43 mostra a mensagem recebida no webmail da conta **thiago@milhomem.org**:

¹⁷ Serviço provido pelo Google para empresas. Maiores informações em:
 <<http://www.google.com/a/help/intl/pt-BR/index.html>>. Acesso em 3 jun. 2009

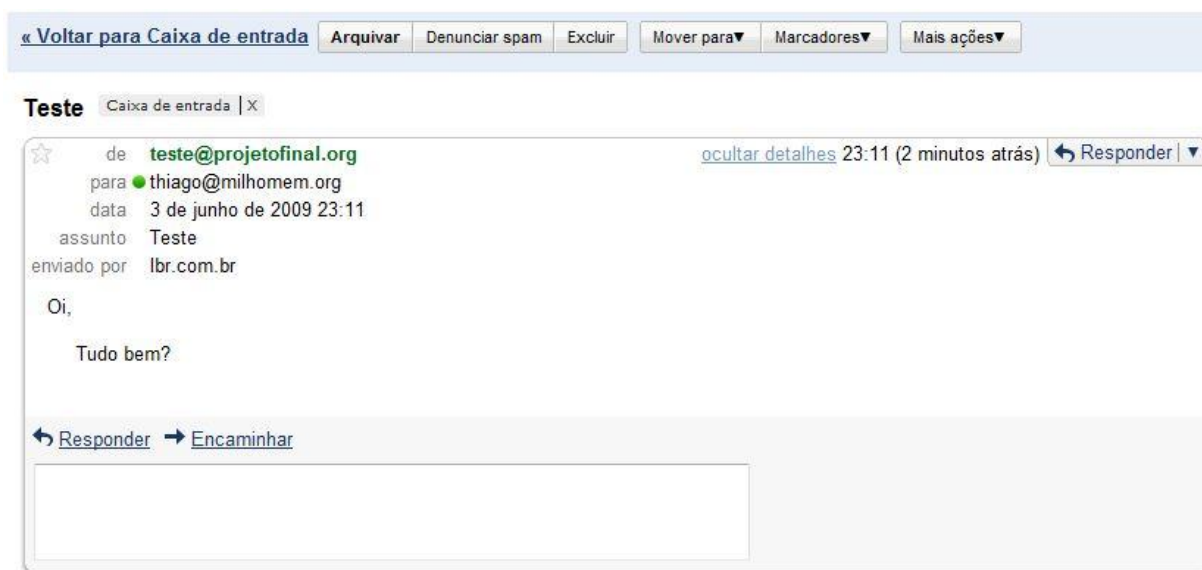


Figura 43: “Teste de envio de mensagem para um domínio na internet.”

Já para o teste de recebimento, a mensagem enviada foi repondida, sendo entregue corretamente para a conta **teste@projetofinal.org**, conforme pode ser observado na figura 44:

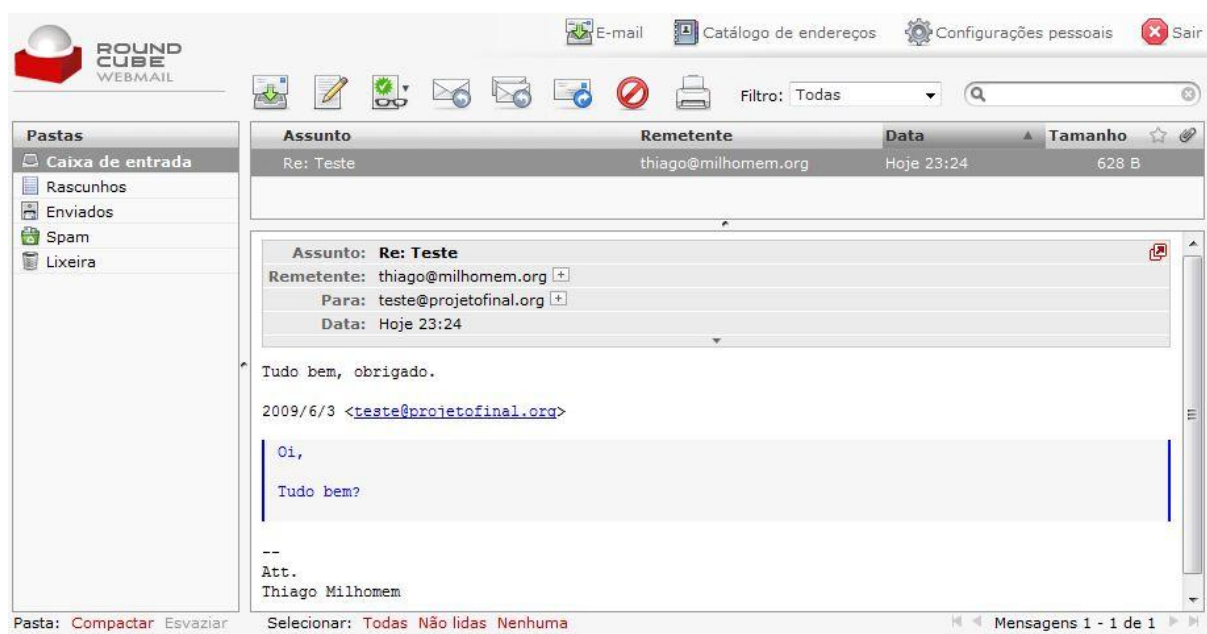


Figura 44: “Teste de recebimento de mensagem de um domínio na internet.”

5.2. Testes dos recursos de segurança

Depois de realizado os testes do funcionamento do servidor, foram testados os recursos de segurança, com a finalidade de averiguar se os mesmos estavam protegendo de forma adequada os dados manipulados.

Os primeiros testes realizados estavam relacionados aos protocolos SSL e TLS, utilizados para manter o sigilo das informações durante a transmissão de dados entre o servidor e um cliente de e-mails. Para isso foi utilizada a ferramenta *Wireshark*, que é um analisador de pacotes. Essa ferramenta foi inserida na topologia de rede criada para os testes, conforme mostra a figura 45:

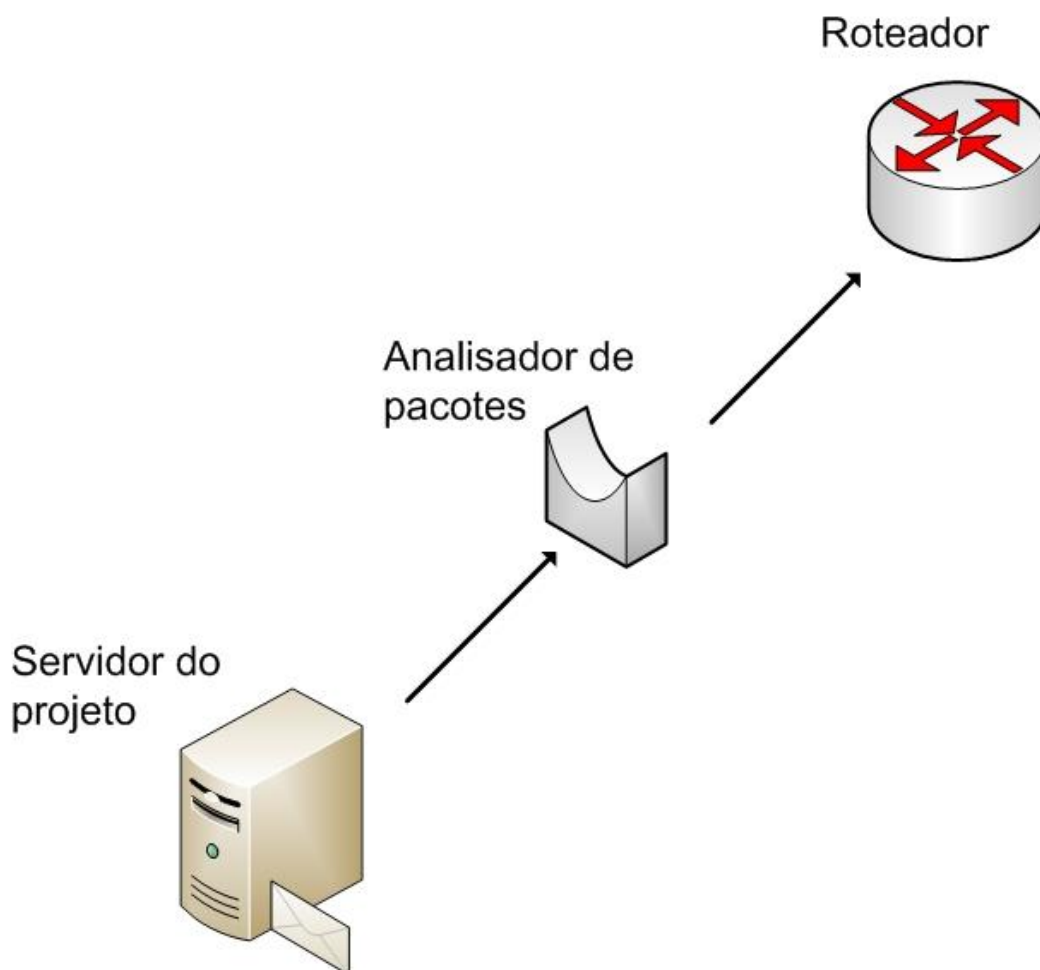
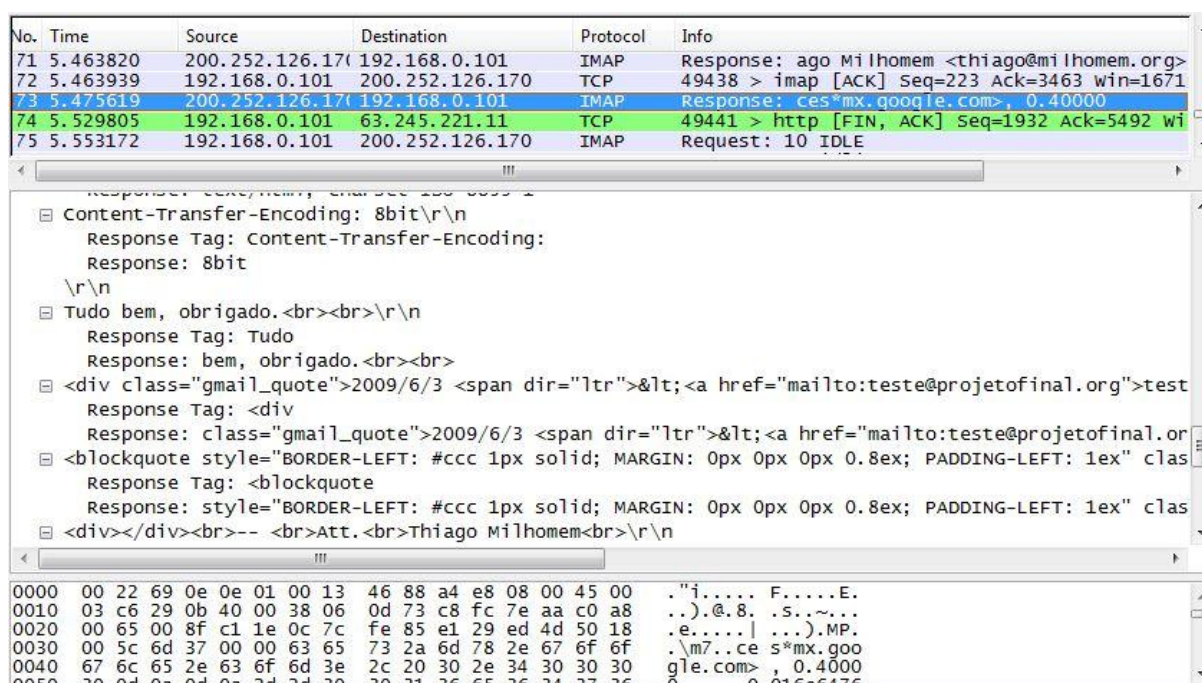


Figura 45: "Topologia de testes com a ferramenta Wireshark."

Ainda na figura 45 pode-se observar que o analisador de pacotes foi inserido entre o servidor do projeto e o roteador. Sendo assim, todo o tráfego de entrada e saída pode ser verificado.

Depois de configurada a ferramenta wireshark, foi acessada a conta **teste@projetofinal.org** em um cliente de e-mail. O protocolo configurado para o acesso às mensagens foi o IMAP com o SSL desabilitado. Durante o acesso à conta foi verificado no analisador que os dados trocados entre o cliente de e-mail e o servidor foram realizados em texto claro. Isso pode ser visto na figura 46:



No.	Time	Source	Destination	Protocol	Info
71	5.463820	200.252.126.170	192.168.0.101	IMAP	Response: ago Milhomem <thiago@milhomem.org>
72	5.463939	192.168.0.101	200.252.126.170	TCP	49438 > imap [ACK] Seq=223 Ack=3463 win=1671
73	5.475619	200.252.126.170	192.168.0.101	IMAP	Response: ces*mx.google.com>, 0.40000
74	5.529805	192.168.0.101	63.245.221.11	TCP	49441 > http [FIN, ACK] Seq=1932 Ack=5492 wi
75	5.553172	192.168.0.101	200.252.126.170	IMAP	Request: 10 IDLE

Content-Transfer-Encoding: 8bit\r\n	
Response Tag: Content-Transfer-Encoding:	
Response: 8bit	
\r\n	
Tudo bem, obrigado. \r\n	
Response Tag: Tudo	
Response: bem, obrigado. 	
<div class="gmail_quote">2009/6/3 <test	
Response Tag: <div	
Response: class="gmail_quote">2009/6/3 <<a href="mailto:teste@projetofinal.or	
<blockquote style="BORDER-LEFT: #ccc 1px solid; MARGIN: 0px 0px 0px 0.8ex; PADDING-LEFT: 1ex" clas	
Response Tag: <blockquote	
Response: style="BORDER-LEFT: #ccc 1px solid; MARGIN: 0px 0px 0px 0.8ex; PADDING-LEFT: 1ex" clas	
<div></div> -- Att. Thiago Milhomem \r\n	

No.	Time	Source	Destination	Protocol	Info
0000	00 22 69 0e 0e 01 00 13	46 88 a4 e8 08 00 45 00	..i.... F....E.		
0010	03 c6 29 0b 40 00 38 06	0d 73 c8 fc 7e aa c0 a8	..).@.8. .s.~...		
0020	00 65 00 8f c1 1e 0c 7c	fe 85 e1 29 ed 4d 50 18	.e.... ...).MP.		
0030	00 5c 6d 37 00 00 63 65	73 2a 6d 78 2e 67 6f 6f	..m7..ce s*mx.goo		
0040	67 6c 65 2e 63 6f 6d 3e	2c 20 30 2e 34 30 30 30	gle.com>, 0.4000		
0050	20 0d 03 0d 03 2d 2d 20	20 21 26 65 26 24 27 26	0 0 01606476		

Figura 46: "Comunicação IMAP em texto claro."

Sem a utilização do SSL, foi possível visualizar claramente o conteúdo da mensagem. Após isso, foi habilitado o SSL e depois realizado o teste de acesso novamente. Dessa vez, foi verificado que todos os dados trocados estavam criptografados, conforme é mostrado na região marcada com a coloração azul, na figura 47:

No.	Time	Source	Destination	Protocol	Info
58	5.670592	192.168.0.101	200.252.126.170	TLSv1	Application Data
59	5.736682	200.252.126.170	192.168.0.101	TLSv1	Application Data
60	5.749407	192.168.0.101	200.252.126.170	TLSv1	Application Data
61	5.826096	63.245.221.10	192.168.0.101	HTTP	HTTP/1.1 200 OK (text/css)
62	5.826224	192.168.0.101	63.245.221.10	TCP	49493 > http [RST, ACK] Seq=968 Ack=1936 w
63	5.885091	200.252.126.170	192.168.0.101	TLSv1	Application Data,
64	5.916717	200.252.126.170	192.168.0.101	TLSv1	Application Data

header length: 20 bytes	
Flags: 0x18 (PSH, ACK)	
window size: 8000 (scaled)	
checksum: 0x0a15 [correct]	
[SEQ/ACK analysis]	
Secure Socket Layer	
TLSv1 Record Layer: Application Data Protocol: imap	
Content Type: Application Data (23)	
Version: TLS 1.0 (0x0301)	
Length: 48	
Encrypted Application Data: E8438A0B9CAAFAC4F249444524A81B497B60E6073FAA6C3E...	

0030	00 7d 0a 15 00 00 17 03	01 00 30 e8 43 8a 0b 9c	..}.....0.C..
0040	aa fa c4 f2 49 44 45 24	a8 1b 49 7b 60 e6 07 3f	...IDES\$.I{...?
0050	aa 6c 3e da 11 dd a5 c3	99 60 f2 2e 07 0f da c7	.l>.....
0060	03 1d 4e b5 f9 60 5f 50	07 99 d3	..N..._P...

Figura 47: “Comunicação IMAP criptografada.”

Os mesmos testes foram realizados para os protocolos POP e SMTP. Ambos geraram os mesmos resultados.

Depois disso foi testado o funcionamento do protocolo HTTPS. Para isso, foi acessada a ferramenta Postfix Admin via navegador de internet. Durante o acesso foi exibida uma página de erro, informando que o servidor acessado não é confiável. A figura 48 mostra a tela de um navegador de internet indicando esse tipo de erro:

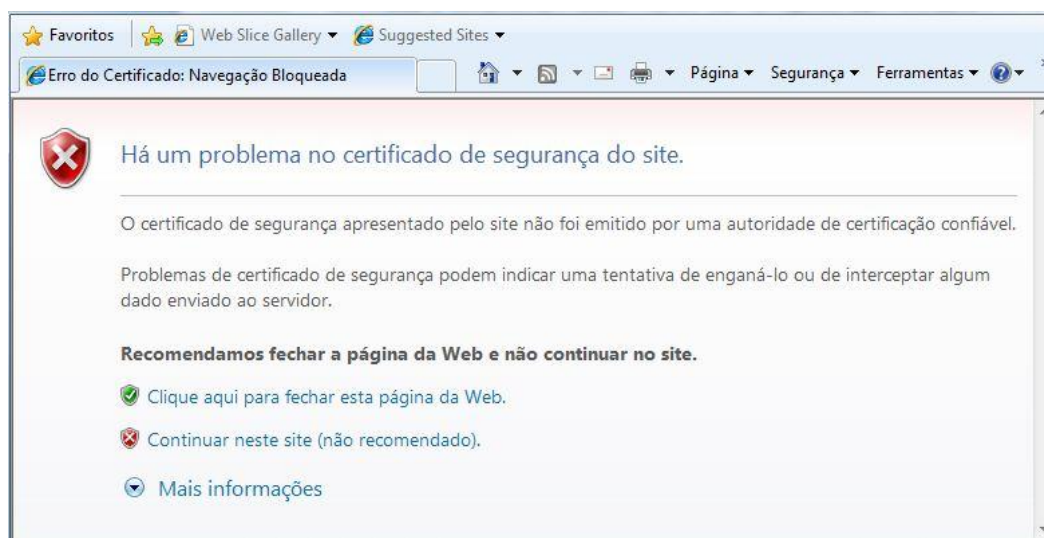


Figura 48: “Erro apresentado pelo navegador de internet.”

Isso ocorre porque os certificados do servidor são auto-assinados e, portanto, não são emitidos por entidades confiáveis. Porém, como o objetivo dos certificados para o projeto é o sigilo dos dados, esse erro pode ser ignorado. Durante os acessos realizados ao endereço da ferramenta Postfix Admin (<https://admin.projetofinal.org>) foi verificado no analisador de pacotes que os dados transmitidos estavam criptografados, conforme pode ser verificado na figura 49, o trecho indicado com a cor azul:

No.	Time	Source	Destination	Protocol	Info
84	21.638616	192.168.0.101	65.55.195.250	TCP	49672 > https [FIN, ACK] Seq=1385 Ack=5403
85	21.662131	65.55.195.250	192.168.0.101	TCP	https > 49673 [ACK] Seq=4553 Ack=1407 win=6
86	21.670126	65.55.195.250	192.168.0.101	TLSv1	Application Data
87	21.670240	192.168.0.101	65.55.195.250	TCP	49673 > https [ACK] Seq=1407 Ack=5403 win=1
88	21.670387	192.168.0.101	65.55.195.250	TCP	49673 > https [FIN, ACK] Seq=1407 Ack=5403

[Next sequence number: 5402	(relative sequence number)]
Acknowledgement number: 1407	(relative ack number)
Header length: 20 bytes	
Flags: 0x19 (FIN, PSH, ACK)	
Window size: 64129	
Checksum: 0x21ba [correct]	
Secure Socket Layer	
TLSv1 Record Layer: Application Data Protocol: http	
Content Type: Application Data (23)	
Version: TLS 1.0 (0x0301)	
Length: 844	
Encrypted Application Data: 0F5A95681C44020A0A98B42C8E82826A66907300CE319337...	

0030	fa 81 21 ba 00 00 17 03	01 03 4c 0f 5a 95 68 1c	...!.....L.Z.h.
0040	44 02 0a 0a 98 b4 2c 8e	82 82 6a 66 90 73 00 ce	D.....jf.s..
0050	31 93 37 c1 fe 7e e5 25	8d bd 1a 60 33 7e b5 e6	1.7...%...3~..
0060	00 92 3a 57 fe b9 1f 04	31 4d 23 67 2a 32 4d 26	..:w...1M#g*2M&
0070	f9 40 33 98 31 c1 67 42	43 29 ef 44 d0 21 43 50	..@3.1.gB C).D.!CP
0080	37 2a 64 ef b3 53 65 b5	26 62 08 dc 1a c2 47 06	..:z..h..C..

Figura 49: “Comunicação HTTP criptografada.”

Depois de realizados todos os testes relacionados aos protocolos SSL e TLS, foi testada a autenticação para o envio de e-mails, provida pelo protocolo SASL. Para a realização do teste, foi configurado o cliente de e-mail da conta **teste@projetofinal.org** sem autenticação para o envio de mensagens. Após isso, foi feita a tentativa de envio de um e-mail para um domínio externo na internet. Durante o processo de envio, foi exibida uma mensagem de erro, informando que a mensagem não pôde ser enviada, conforme é mostrado na figura 50:

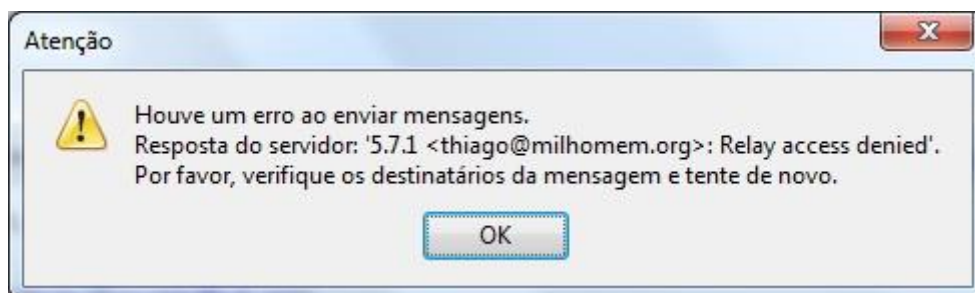


Figura 50: “Erro na tentativa de envio de mensagem sem a autenticação habilitada.”

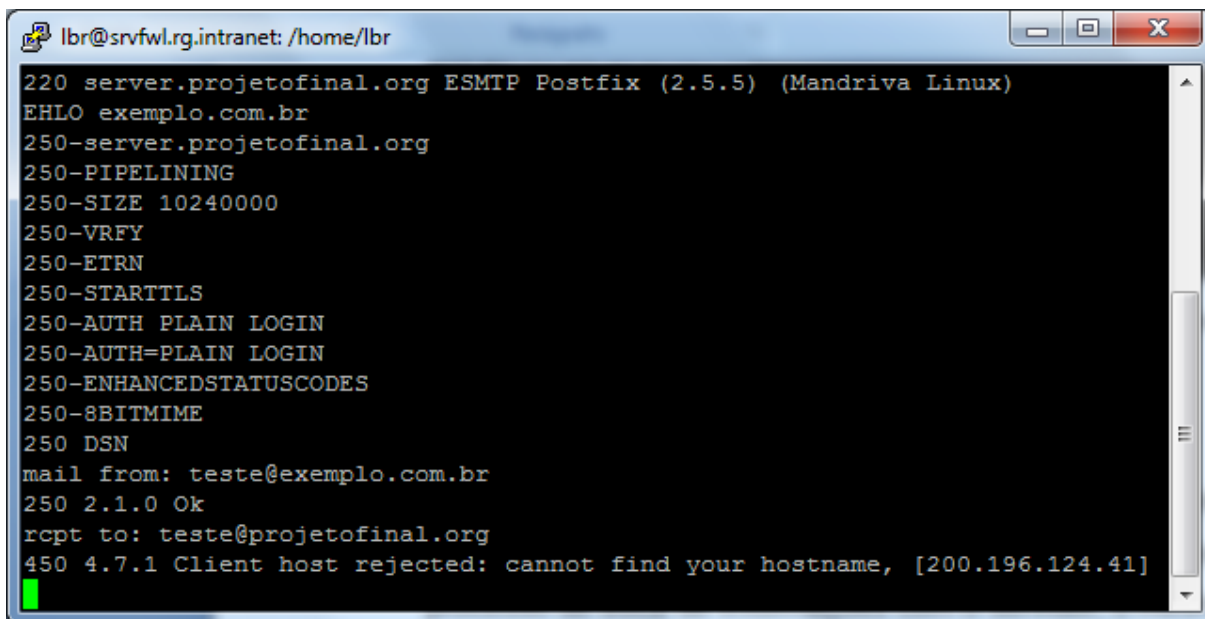
Os próximos testes diziam a respeito das restrições de mensagens configuradas no software Postfix. Como essas restrições são aplicadas durante os comandos SMTP executados durante a conversação com o servidor, os testes foram realizados através do acesso ao serviço via telnet. No primeiro teste foi informado durante a conversação com o servidor um remetente de um domínio não registrado na internet, não possuindo, portanto, registros DNS associados a ele. Neste caso o servidor rejeitou o recebimento da mensagem, pois não havia encontrado um registro DNS para o domínio do remetente, conforme exibido na figura 51:

```

220 server.projetofinal.org ESMTP Postfix (2.5.5) (Mandriva Linux)
EHLO exemplo.net.br
250-server.projetofinal.org
250-PIPELINING
250-SIZE 10240000
250-URFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: teste@exemplo.net.br
250 2.1.0 Ok
rcpt to: teste@projetofinal.org
450 4.1.8 <teste@exemplo.net.br>: Sender address rejected: Domain not found
  
```

Figura 51: “Bloqueio de remetente de domínio desconhecido.”

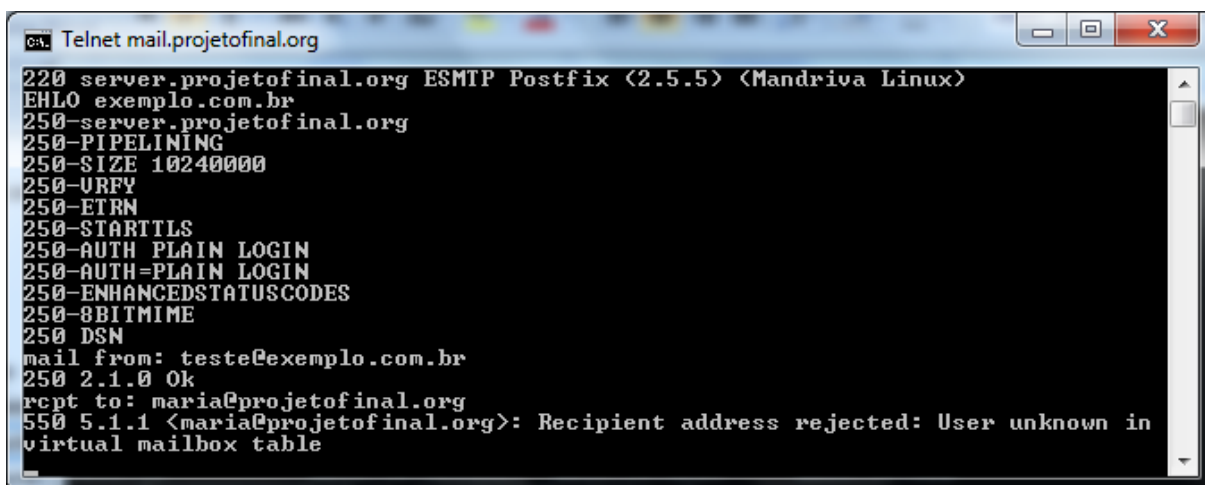
No segundo teste foi feita a conexão no servidor do projeto através de um endereço IP da internet que não possuía um registro DNS reverso associado. Durante o processo de troca de informações com o servidor, o mesmo rejeitou a finalização da conversação, e consequentemente, o recebimento da mensagem, pois não conseguiu encontrar o registro DNS reverso do endereço IP utilizado. A rejeição ocorrida pode ser verificada na figura 52:



```
lbr@srvfwl.rg.intranet: /home/lbr
220 server.projetofinal.org ESMTP Postfix (2.5.5) (Mandriva Linux)
EHLO exemplo.com.br
250-server.projetofinal.org
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: teste@exemplo.com.br
250 2.1.0 Ok
rcpt to: teste@projetofinal.org
450 4.7.1 Client host rejected: cannot find your hostname, [200.196.124.41]
```

Figura 52: “Bloqueio de remetente de domínio de DNS reverso configurado.”

O terceiro e último teste relacionado às restrições do Postfix foi a tentativa de envio de uma mensagem para um destinatário inexistente na base de dados do servidor. Durante o processo de conversação o servidor rejeitou a mensagem, pois alegava que o destinatário não existia em sua base de dados, conforme pode ser verificado na figura 53:

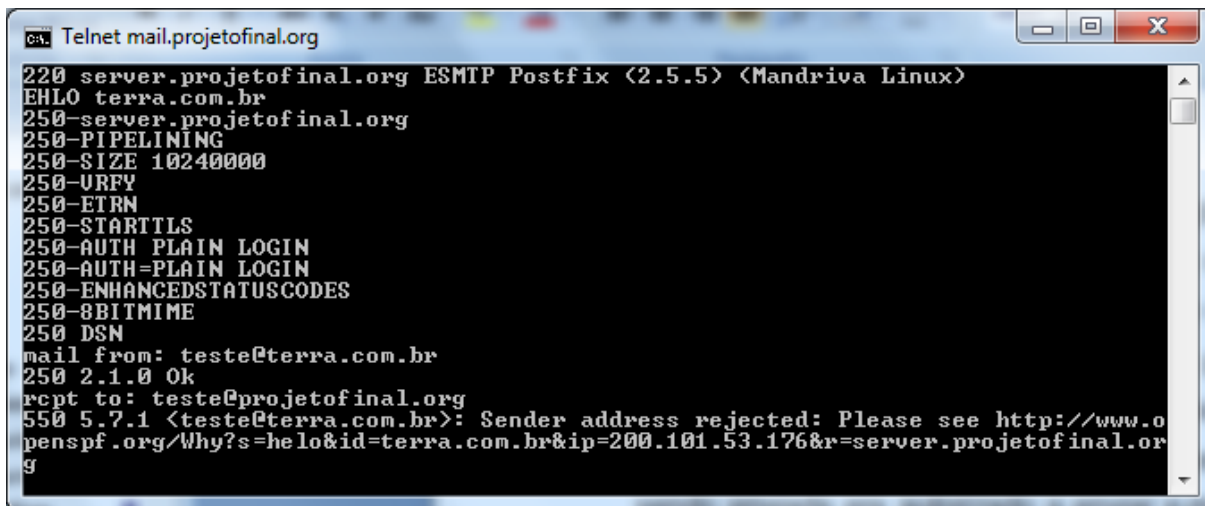


```
Telnet mail.projetofinal.org
220 server.projetofinal.org ESMTP Postfix (2.5.5) (Mandriva Linux)
EHLO exemplo.com.br
250-server.projetofinal.org
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: teste@exemplo.com.br
250 2.1.0 Ok
rcpt to: maria@projetofinal.org
550 5.1.1 <maria@projetofinal.org>: Recipient address rejected: User unknown in
virtual mailbox table
```

Figura 53: “Mensagem descartada devido a destinatário inexistente na base de dados.”

Depois dos testes acerca das restrições do Postfix, foram testados os filtros SPF e Postgrey. Para o teste do SPF foi realizada a tentativa de envio de uma mensagem se passando como um usuário do domínio **terra.com.br**, porém de um

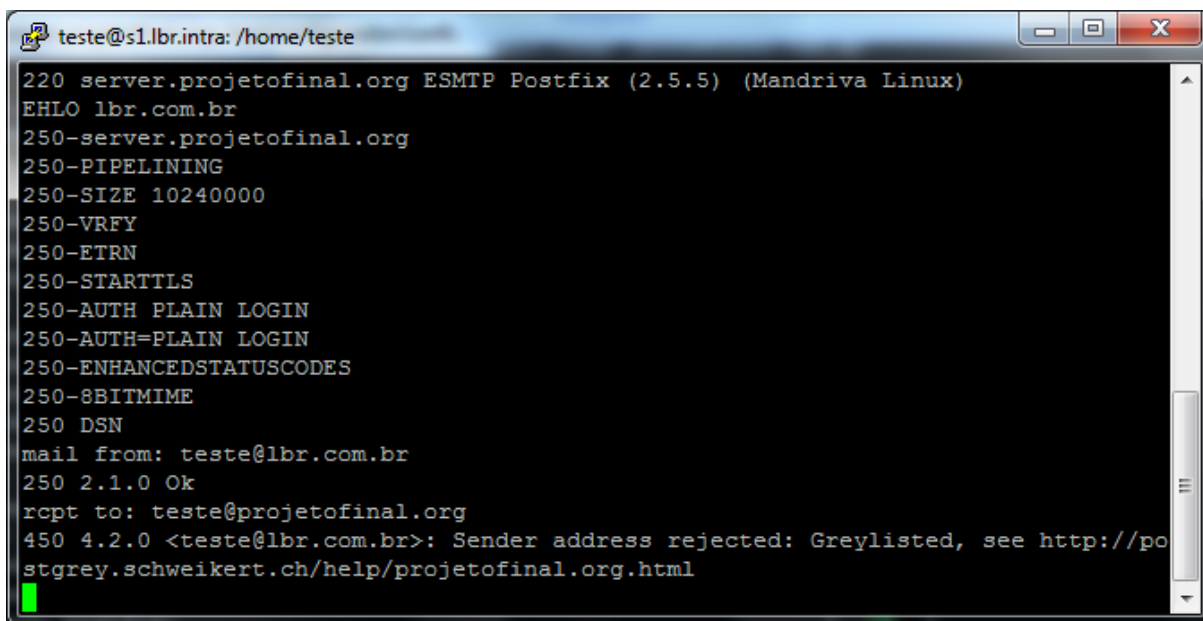
endereço IP não autorizado pelo mesmo para o envio de mensagens. Após as verificações do Postfix durante a fase inicial da conversação, a mensagem foi analisada pelo filtro SPF, que verificou se o endereço IP no qual a mensagem estava sendo enviada era autorizado a enviar e-mail em nome do domínio **terra.com.br**. Como esse endereço não era autorizado, o remetente foi rejeitado, conforme é exibido na figura 54:



```
Telnet mail.projetofinal.org
220 server.projetofinal.org ESMTP Postfix (2.5.5) (Mandriva Linux)
EHLO terra.com.br
250-server.projetofinal.org
250-PIPELINING
250-SIZE 10240000
250-URFU
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: teste@terra.com.br
250 2.1.0 Ok
rcpt to: teste@projetofinal.org
550 5.7.1 <teste@terra.com.br>: Sender address rejected: Please see http://www.o
penspf.org/Why?s=hello&id=terra.com.br&ip=200.101.53.176&r=server.projetofinal.or
g
```

Figura 54: “Mensagem bloqueada pelo filtro SPF.”

Já para o teste do filtro Postgrey, foi enviada uma mensagem de forma com que a mesma atendesse os requisitos das restrições do Postfix e do filtro SPF. Após as verificações dos mesmos, a mensagem foi rejeitada temporariamente, aplicando corretamente o conceito de *greylisting* - já abordado no tópico 4.2.4.2. A partir da figura 55 é possível verificar o bloqueio da mensagem pelo filtro Postgrey:



```

teste@s1.lbr.intra: /home/teste
220 server.projetofinal.org ESMTF Postfix (2.5.5) (Mandriva Linux)
EHLO lbr.com.br
250-server.projetofinal.org
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: teste@lbr.com.br
250 2.1.0 Ok
rcpt to: teste@projetofinal.org
450 4.2.0 <teste@lbr.com.br>: Sender address rejected: Greylisted, see http://postgrey.schweikert.ch/help/projetofinal.org.html

```

Figura 55: “Mensagem bloqueada pelo filtro Postgrey.”

Por fim, os últimos testes realizados estiveram relacionados ao software de antispam DSPAM. Para que uma mensagem fosse analisada por esse software, ela precisou atender a todos os requisitos necessários dos filtros abordados anteriormente. Com a finalidade de testar a reaprendizagem do DSPAM, foram enviadas diversas mensagens a partir de um servidor de e-mail configurado corretamente com um domínio válido na internet. Em seus conteúdos foram inseridas palavras que geralmente caracterizam os spams. Devido ao filtro do antispam ainda não estar treinado, todas as mensagens recebidas foram classificadas como legítimas. A partir da figura 56 é possível verificar, através da interface web do software DSPAM, a ocorrência da classificação “Good” para todas as mensagens recebidas:

Performance Preferences Alerts Quarantine (Empty) Analysis **History**

The messages that have been processed by the filter are shown below. The most recent messages are shown first. Use the retrain options to correct errors and deliver any false positives that are still in your quarantine.

Retrain Checked

[1]

Type	Retrain	Day/Time	From	Subject	Additional Info
Good	<input type="checkbox"/> As Spam	Fri 9:59p	"Lowell Savage" <aestheticallyn79@kuppelprojekt...	Get a required degree.	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:59p	"Bvlgari Watches" <zhenyanc@basf.com>	2010 Swiss Rolex	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:59p	"Collin Pritchard" <spattingeh24@sap-rfid.com&g...	Let Accai Berry improve your health	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:59p	<sag@europarl.eu.int>	Software license!	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:59p	"Mai Sandoval" <luz.marzano@gmail.com>	Affordable brand name watches	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:58p	"Lowell Savage" <aestheticallyn79@kuppelprojekt...	Get a required degree.	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:58p	"Bvlgari Watches" <zhenyanc@basf.com>	2010 Swiss Rolex	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:58p	"Collin Pritchard" <spattingeh24@sap-rfid.com&g...	Let Accai Berry improve your health	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:58p	<sag@europarl.eu.int>	Software license!	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:58p	"Mai Sandoval" <luz.marzano@gmail.com>	Affordable brand name watches	Delivered
Good	<input type="checkbox"/> As Spam	Fri 9:58p	"Lowell Savage" <aestheticallyn79@kuppelprojekt...	Get a required degree.	Delivered

Figura 56: "Filtro do software DSPAM sem treinamento."

Para realizar a reaprendizagem dessas mensagens, classificando-as como spams, foi utilizada a própria interface web do software de antispam. Após isso, novas mensagens com conteúdo semelhante foram enviadas, e essas foram classificadas como spams, conforme podem ser observadas as ocorrências da classificação "SPAM" na figura 57:

Performance Preferences Alerts Quarantine (Empty) Analysis **History**

The messages that have been processed by the filter are shown below. The most recent messages are shown first. Use the retrain options to correct errors and deliver any false positives that are still in your quarantine.

Retrain Checked

[1]

Type	Retrain	Day/Time	From	Subject	Additional Info
SPAM	<input type="checkbox"/> As Innocent	Fri 10:02p	"Lowell Savage" <aestheticallyn79@kuppelprojekt...	Get a required degree.	Quarantined
SPAM	<input type="checkbox"/> As Innocent	Fri 10:02p	"Bvlgari Watches" <zhenyanc@basf.com>	2010 Swiss Rolex	Quarantined
SPAM	<input type="checkbox"/> As Innocent	Fri 10:01p	"Collin Pritchard" <spattingeh24@sap-rfid.com&g...	Let Accai Berry improve your health	Quarantined
SPAM	<input type="checkbox"/> As Innocent	Fri 10:01p	<sag@europarl.eu.int>	Software license!	Quarantined
SPAM	<input type="checkbox"/> As Innocent	Fri 10:01p	"Mai Sandoval" <luz.marzano@gmail.com>	Affordable brand name watches	Quarantined
Miss	<input type="checkbox"/> Retrained (Undo)	Fri 9:59p	"Lowell Savage" <aestheticallyn79@kuppelprojekt...	Get a required degree.	Delivered
Miss	<input type="checkbox"/> Retrained (Undo)	Fri 9:59p	"Bvlgari Watches" <zhenyanc@basf.com>	2010 Swiss Rolex	Delivered
Miss	<input type="checkbox"/> Retrained (Undo)	Fri 9:59p	"Collin Pritchard" <spattingeh24@sap-rfid.com&g...	Let Accai Berry improve your health	Delivered
Miss	<input type="checkbox"/> Retrained (Undo)	Fri 9:59p	<sag@europarl.eu.int>	Software license!	Delivered

Figura 57: "Filtro do software DSPAM já treinado."

5.3. Resultados da remasterização

O que se esperou alcançar com o processo de remasterização da imagem de instalação do sistema foi tornar o processo de instalação e configuração do servidor mais rápido. Além disso, foi esperado minimizar a possibilidade de erros de configuração dos serviços devido à grande quantidade de arquivos que precisavam ser editados.

Durante o teste da instalação do servidor através da imagem de instalação remasterizada, foi observado que diversas etapas foram executadas automaticamente. Entre elas destacam-se:

- Criação de dez partições no disco rígido do servidor, cada uma com propósito específico;
- Formatação das dez partições com o sistema de arquivos XFS;
- Seleção dos pacotes básicos necessários para a inicialização correta do servidor;
- Criação do usuário comum projeto, para ser utilizado no acesso remoto ao servidor.

Com isso, a instalação do servidor tornou-se prática, pois não foi necessário realizar as configurações mais complexas do processo de instalação. Além disso, tempo levado para instalar o servidor reduziu consideravelmente. A partir da tabela 1 é possível verificar os tempos levados para a instalação - tanto para a imagem original do sistema, como para a imagem remasterizada - em dez testes distintos:

Tabela 1: “Tempos ocorridos durante os testes de instalação.”

Teste	Instalação imagem original	Instalação imagem remasterizada
1	13,38 minutos	4,45 minutos
2	13,20 minutos	4,46 minutos

3	13,25 minutos	4,42 minutos
4	12,45 minutos	4,48 minutos
5	13,55 minutos	4,50 minutos
6	14,10 minutos	4,45 minutos
7	13,42 minutos	4,47 minutos
8	12,30 minutos	4,46 minutos
9	12,05 minutos	4,46 minutos
10	13,00 minutos	4,44 minutos
média:	13,07 minutos	4,46 minutos

Ainda na tabela 1, observa-se que o tempo médio levado para a instalação do servidor utilizando a imagem remasterizada foi menor que a metade do tempo médio levado para se realizar uma instalação.

Após os testes realizados acerca da instalação do servidor, foram realizados testes sobre a configuração do mesmo. Após a primeira inicialização do servidor instalado com a imagem remasterizada, o script **bemvindo.sh** é executado automaticamente. Através dele são realizadas as seguintes configurações:

- Instalação de todos os softwares necessários ao funcionamento do servidor;
- Alteração da senha do usuário comum projeto, pois o mesmo foi configurado durante a instalação de forma não-interativa;
- Alteração da senha do usuário administrativo root, pelo mesmo motivo do item anterior;
- Realização das configurações de rede através da execução do utilitário da distribuição Mandriva **drakconnect**;

Após a realização das configurações iniciais, foi mostrada uma tela informando a existência do arquivo **Leia-me.txt**, situado no diretório **/opt/configurador**. Ele

contém informações sobre como proceder para finalizar a instalação do servidor, além dos pré-requisitos para o seu correto funcionamento, como por exemplo, as configurações de DNS. Após a leitura do arquivo, foi editado o arquivo de configuração `variables`, situado em **`/opt/configurador/bin`**. Nesse arquivo foram informados os dados necessários para a geração dos certificados auto-assinados, além do nome domínio principal do servidor. Após isso, foi executado o seguinte comando:

```
# configurador.sh full
```

Com isso, todo o servidor foi configurado a partir dos dados informados no arquivo **`variables`**.

É importante notar que a complexidade da configuração do servidor foi drasticamente reduzida, visto que foi necessário editar apenas um arquivo de configuração, além de seguir os passos descritos no script de configuração inicial **`bemvindo.sh`**. Para a implementação do servidor de forma manual, é necessário a edição de mais de vinte e cinco arquivos de configuração, entre eles os arquivos **`main.cf`** e **`dovecot.conf`**, responsáveis por grande parte das configurações dos softwares Postfix e Dovecot, respectivamente.

6. CONCLUSÃO

O projeto conseguiu atingir o seu objetivo, que é uma solução para servidores de e-mail que possui facilidade na sua instalação e configuração, além de possuir recursos de segurança e ferramentas que facilitam a administração das contas de e-mail.

Através dos testes realizados, foi possível verificar que a criação das contas de e-mail na base de dados ocorreu de forma correta e o envio e recebimento de mensagens - tanto via cliente de e-mail como via webmail - funcionaram sem erros.

Além disso, os filtros SPF e Postgrey funcionaram corretamente. As mensagens que não atendiam os requisitos necessários determinados por esses filtros foram descartadas. Os filtros do software Postfix também bloquearam as mensagens que não atendiam os seus requisitos para entrega.

O software de antispam DSPAM conseguiu acertar a classificação das mensagens, após certo período de treinamento. Os mecanismos para a reaprendizagem das mensagens também funcionaram corretamente. O DSPAM conseguiu bloquear as mensagens que possuíam vírus em seus anexos, devido à sua integração com o software de antivírus ClamAV.

O uso dos protocolos POP, IMAP e SMTP com a criptografia provida pelo SSL e o TLS garantiram que os dados transmitidos durante a comunicação entre o cliente de e-mail e o servidor estivessem seguros.

Através dos dados mostrados no capítulo 5 foi possível concluir que o processo de automatização da instalação do sistema criado na solução reduziu sensivelmente o tempo levado para se instalar o sistema. Além disso, os scripts criados para automatizar o processo de configuração do servidor tornaram a implementação da solução rápida e fácil de realizar, podendo ser realizada por administradores que possuem uma experiência reduzida no que diz respeito aos serviços de e-mail.

Uma das dificuldades encontradas durante a implementação do projeto foi a integração das ferramentas de segurança ao servidor de e-mail. Encontrar uma forma adequada e eficiente de analisar a mensagem antes da mesma ser entregue foi um verdadeiro desafio. Outra dificuldade encontrada foi a remasterização da imagem de instalação da distribuição Mandriva Linux. Existe pouca documentação sobre o assunto. Por fim, hospedar o servidor na internet para realizar os devidos testes gerou diversos problemas, entre eles podem se destacar os gastos gerados para se obter um link de internet empresarial e o registro de um domínio na internet.

Algumas sugestões para projetos futuros são:

- Criar uma solução para servidores de e-mail com a base de dados LDAP, ao invés de banco de dados;
- Realizar uma análise do tipo de algoritmo de criptografia mais adequado para os serviços de e-mail;
- Criar ferramentas que automatizem o processo de instalação e configuração em outras linguagens de programação, como por exemplo, o PHP.

REFERÊNCIAS BIBLIOGRÁFICAS

CGI.BR, Comitê Gestor da Internet no Brasil. **Filtros de Conteúdo**. Disponível em: <http://www.antispam.br/admin/filtros/>. Acesso em 19 mai 2009.

CGI.BR, Comitê Gestor da Internet no Brasil. **Greylisting**. Disponível em: <http://www.antispam.br/admin/greylisting/>. Acesso em 16 mai 2009.

CGI.BR, Comitê Gestor da Internet no Brasil. **O que é spam?** Disponível em: <http://www.antispam.br/conceito/>. Acesso em 16 mai 2009.

CGI.BR, Comitê Gestor da Internet no Brasil. **SPF – Sender Policy Framework**. Disponível em: <http://www.antispam.br/admin/spf/>. Acesso em 16 mai 2009.

DENT, Kyle D. **Postfix: The Definitive Guide** – 1st ed. – California, EUA. O'Reilly Media:2003

GUPTA, Meeta; PARIHAR, Mridula; SCRIMGER, Rod; LASALLE, Paul. **TCP/IP Bible** – 1st ed. – New Jersey, EUA. Wiley:2001

HEWLETT-PACKARD. O que é virtualização e o que ela pode fazer para a minha empresa? Disponível em: http://www.hp.com/latam/br/pyme/solucoes/apr_solucoes_01.html. Acesso em 26 mar. 2009

MICROSOFT. **Virtualização Microsoft – Virtualização de servidor**. Disponível em: <http://www.microsoft.com/brasil/servidores/virtualizacao/solution-tech-server.msp>. Acesso em 26 mar. 2009.

NAKAMURA, Emilio Tissato; GEUS, Paulo Lício de. **Segurança de redes em ambientes corporativos**. 2 ed. São Paulo, Futura: 2003

SILICON GRAPHICS. **Open Source XFS for Linux**. Disponível em: <http://oss.sgi.com/projects/xfs/datasheet.pdf>. Acesso em 17 mai 2009.

WOOD, David. **Programming Internet Email**. 1st ed. California, EUA. O'Reilly Media:1999

WORSLEY, John; DRAKE Joshua. **Practical PostgreSQL** – 1st ed. – California, EUA. O'Reilly Media:2001

Apêndice A - Arquivos de configuração do software postfix:

/etc/postfix/main.cf:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração main.cf

readme_directory = /usr/share/doc/postfix/README_FILES
html_directory = /usr/share/doc/postfix/html
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
command_directory = /usr/sbin
manpage_directory = /usr/share/man
daemon_directory = /usr/lib/postfix
newaliases_path = /usr/bin/newaliases
mailq_path = /usr/bin/mailq
queue_directory = /var/spool/postfix
mail_owner = postfix
mynetworks_style = host
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version) (Mandriva Linux)
unknown_local_recipient_reject_code = 450
smtp-filter_destination_concurrency_limit = 2
lmtp-filter_destination_concurrency_limit = 2
data_directory = /var/lib/postfix

inet_interfaces = all
mynetworks = 127.0.0.1/32
mydomain = projetofinal.org
myhostname = servidor.projetofinal.org

smtpd_use_tls = yes
smtpd_tls_cert_file = /etc/ssl/projeto/certs/email.crt
smtpd_tls_key_file = /etc/ssl/projeto/private/email.key

virtual_transport = virtual
virtual_mailbox_base = /home/vmail
virtual_mailbox_domains = proxy:pgsql:/etc/postfix/pgsql/virtual-domains.cf
virtual_alias_maps = proxy:pgsql:/etc/postfix/pgsql/virtual-alias-maps.cf
virtual_mailbox_maps = proxy:pgsql:/etc/postfix/pgsql/virtual-mailbox-maps.cf
virtual_uid_maps = static:500
virtual_gid_maps = static:500
transport_maps = hash:/etc/postfix/transport
local_recipient_maps = $transport_maps
virtual_create_maildirsize = yes
virtual_maildir_extended = yes
virtual_mailbox_limit_maps = proxy:pgsql:/etc/postfix/pgsql/virtual-mailbox-limit-maps.cf
virtual_mailbox_limit_override = yes
virtual_maildir_limit_message = "The user you are trying to reach is over quota."
virtual_overquota_bounce = yes

proxy_read_maps      =      $local_recipient_maps      $mydestination      $virtual_alias_maps
$virtual_alias_domains  $virtual_mailbox_maps  $virtual_mailbox_domains  $relay_recipient_maps
$relay_domains        $canonical_maps        $sender_canonical_maps        $recipient_canonical_maps

```

```
$relocated_maps $transport_maps $mynetworks $sender_bcc_maps $recipient_bcc_maps
$smtp_generic_maps $lmtp_generic_maps $virtual_mailbox_limit_maps
```

```
smtpd_helo_required = yes
smtpd_reject_unlisted_recipient = yes
smtpd_helo_restrictions =
    permit_mynetworks,
    reject_invalid_hostname,
    reject_non_fqdn_hostname
```

```
smtpd_sender_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination,
    check_policy_service unix:private/policy,
    check_policy_service inet:127.0.0.1:10031,
    reject_unknown_sender_domain,
    reject_invalid_hostname,
    reject_non_fqdn_sender,
    reject_unknown_recipient_domain
```

```
smtpd_client_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unknown_client,
    check_client_access pcre:/etc/postfix/dspam_filter_access
    reject_rbl_client bl.spamcop.net,
    reject_rbl_client dnsbl.njabl.org,
    reject_rbl_client list.dsbl.org,
    reject_rbl_client sbl.spamhaus.org
```

```
smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    check_recipient_access hash:/etc/postfix/access,
    permit_auth_destination,
    permit_mynetworks,
    reject_unauth_destination,
    reject_non_fqdn_sender,
    reject_non_fqdn_recipient
```

```
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
```

/etc/postfix/master.cf:

```
##Projeto Final
##Thiago de Almeida Milhomem
## Arquivo de configuração master.cf
```

```
smtp inet n - y - - smtpd
```

```
dspam unix - n n - 10 pipe
flags=Ru user=dspam argv=/usr/bin/dspam
--deliver=innocent --user ${recipient}
```

```
dspam-retrain unix - n n - 2 pipe
flags=Ru user=dspam argv=/usr/bin/dspam --client --source=error --class=${nexthop} --user
${sender}
retry unix - - y - - error
```

```
policy unix - n n - - spawn
user=nobody argv=/usr/bin/perl /usr/lib/postfix/postfix-policyd-spf-perl
```

```
localhost:10026 inet n - n - - smtpd
-o content_filter=
-o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
-o smtpd_helo_restrictions=
-o smtpd_client_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

```
pickup fifo n - y 60 1 pickup
-o content_filter=
-o receive_override_options=
cleanupunix n - y - 0 cleanup
qmgr fifo n - y 300 1 qmgr
tlsmgr unix - - y 1000? 1 tlsmgr
rewrite unix - - y - - trivial-rewrite
bounce unix - - y - 0 bounce
defer unix - - y - 0 bounce
trace unix - - y - 0 bounce
verify unix - - y - 1 verify
flush unix n - y 1000? 0 flush
proxymap unix - - n - - proxymap
proxywrite unix - - n - 1 proxymap
smtp unix - - y - - smtp
relay unix - - y - - smtp
-o smtp_fallback_relay=
showq unix n - y - - showq
error unix - - y - - error
retry unix - - y - - error
discard unix - - y - - discard
local unix - n n - - local
virtual unix - n n - - virtual
lmtp unix - - y - - lmtp
anvil unix - - y - 1 anvil
```

/etc/postfix/pgsql/virtual-alias-maps.cf:

```
##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo para pesquisa dos aliases na base de dados
user = postfix
password = 3ed4rf
dbname = postfix
hosts = localhost
query = SELECT goto FROM alias WHERE address='%s' AND active = true
```


/etc/postfix/pgsql/virtual-domains.cf:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo para pesquisa dos domínios na base de dados
user      = postfix
password  = 3ed4rf
dbname    = postfix
hosts     = localhost
query = SELECT domain FROM domain WHERE domain='%s' AND backupmx = false AND active =
true

```

/etc/postfix/pgsql/virtual-mailbox-limit-maps.cf:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo para pesquisa das quotas na base de dados
user = postfix
password = 3ed4rf
hosts = localhost
dbname = postfix
query = SELECT quota FROM mailbox WHERE username = '%s'

```

/etc/postfix/pgsql/virtual-mailbox-maps.cf:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo para pesquisa das contas de e-mail na base de dados
user      = postfix
password  = 3ed4rf
dbname    = postfix
hosts     = localhost
query = SELECT maildir FROM mailbox WHERE username='%s' AND active = true

```

Apêndice B - Arquivos de configuração do software dovecot**/etc/dovecot.conf:**

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração do Dovecot

protocols = imap imaps pop3 pop3s
disable_plaintext_auth = no
syslog_facility = mail

ssl_disable = no
ssl_cert_file = /etc/ssl/projeto/certs/email.crt
ssl_key_file = /etc/ssl/projeto/private/email.key

```

```

login_max_connections = 256
login_greeting = Dovecot ready.

mail_location = maildir:/home/vmail/%d/%n
mail_uid = 500
mail_gid = 500
first_valid_uid = 500
first_valid_gid = 500

protocol pop3 {
    pop3_uidl_format = %08Xu%08Xv
}

protocol lda {
    postmaster_address = postmaster@projetofinal.org
    sendmail_path = /usr/lib/sendmail
    auth_socket_path = /var/run/dovecot/auth-master
}

auth default {
    mechanisms = plain login
    passdb sql {
        args = /etc/dovecot-pgsql.conf
    }
    userdb sql {
        args = /etc/dovecot-pgsql.conf
    }
    user = root
    socket listen {
        master {
            path = /var/run/dovecot/auth-master
            mode = 0666
            user = dspam
            group = dspam
        }
        client {
            path = /var/spool/postfix/private/auth
            mode = 0660
            user = postfix
            group = postfix
        }
    }
}

```

/etc/dovecot-pgsql.conf:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração do Dovecot para conexão na base de dados

driver = pgsql
connect = host=localhost dbname=postfix user=postfix password=3ed4rf
default_pass_scheme = CRYPT
user_query = SELECT username FROM mailbox WHERE username='%u';
password_query = SELECT password FROM mailbox WHERE username='%u';

```

Apêndice C - Arquivos de configuração do software apache

/etc/httpd/conf/vhosts.d/postfixadmin.conf:

```
##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração para o Virtual Host admin.projetofinal.org

<Directory /var/www/postfixadmin>
    Options Indexes FollowSymLinks
    AllowOverride Limit
    order allow,deny
    allow from all
</Directory>

<VirtualHost *:443>
    ServerAdmin webmaster@postfixadmin
    DocumentRoot /var/www/postfixadmin
    ServerName admin.projetofinal.org
    ErrorLog /var/log/httpd/postfixadmin-ssl-error_log
    CustomLog /var/log/httpd/postfixadmin-ssl-access_log common

    SSLEngine on
    SSLCertificateFile /etc/ssl/projeto/certs/postfixadmin.crt
    SSLCertificateKeyFile /etc/ssl/projeto/private/postfixadmin.key

#Dspam cgi

Alias /dspam /usr/share/dspam/cgi-bin
<Directory /usr/share/dspam/cgi-bin/>

    Options ExecCGI
    AllowOverride Limit AuthConfig
    DirectoryIndex dspam.cgi
    Order deny,allow
    Deny from all
    allow from all
    Auth_IMAP_Enabled on
    AuthName "Authorization required"
    AuthType Basic
    Require valid-user
    Auth_IMAP_Authoritative on
    Auth_IMAP_Server localhost
    Auth_IMAP_Port 143
    Auth_IMAP_Log on
</Directory>
</VirtualHost>
```

/etc/httpd/conf/vhosts.d/roundcubemail.conf:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração para o Virtual Host webmail.projetofinal.org
<Directory /var/www/roundcubemail>
    Options Indexes FollowSymLinks
    AllowOverride All
    order allow,deny
    allow from all
</Directory>

NameVirtualHost *:443
<VirtualHost *:443>
    ServerAdmin webmaster@roundcubemail
    DocumentRoot /var/www/roundcubemail
    ServerName webmail.projetofinal.org
    ErrorLog /var/log/httpd/roundcubemail-ssl-error_log
    CustomLog /var/log/httpd/roundcubemail-ssl-access_log common

    SSLEngine on
    SSLCertificateFile /etc/ssl/projeto/certs/webmail.crt
    SSLCertificateKeyFile /etc/ssl/projeto/private/webmail.key

</VirtualHost>

```

Apêndice D - Arquivos de configuração dos softwares dspam e clamav**/etc/dspam.conf:**

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração do software DSPAM

Home /var/lib/dspam
StorageDriver /usr/lib/dspam/libpgsql_drv.so

TrustedDeliveryAgent "/usr/sbin/sendmail"
DeliveryHost      127.0.0.1
DeliveryPort      10026
DeliveryIdent     localhost
DeliveryProto     SMTP

QuarantineAgent    "/usr/lib/dovecot/deliver -d %u -m Spam"
OnFail error

Trust root
Trust mail
Trust mailnull
Trust smmsp
Trust daemon
Trust dspam
Trust apache

```

Trust postfix
Trust train

TrainingMode teft
TestConditionalTraining on
Feature whitelist
Algorithm graham burton
Tokenizer chain
PValue bcr
WebStats on

Preference "spamAction=quarantine"
Preference "signatureLocation=message" # 'message' or 'headers'
Preference "showFactors=on"

AllowOverride trainingMode
AllowOverride spamAction spamSubject
AllowOverride statisticalSedation
AllowOverride enableBNR
AllowOverride enableWhitelist
AllowOverride signatureLocation
AllowOverride showFactors
AllowOverride optIn optOut
AllowOverride whitelistThreshold
AllowOverride localstore
AllowOverride WebStats

PgSQLServer 127.0.0.1
PgSQLPort 5432
PgSQLUser dspam
PgSQLPass 3ed4rf
PgSQLDb dspam
PgSQLConnectionCache 3
PgSQLUIDInSignature on

HashRecMax 98317
HashAutoExtend on
HashMaxExtents 0
HashExtentSize 49157
HashPctIncrease 10
HashMaxSeek 10
HashConnectionCache 10

Notifications off

PurgeSignatures 14 # Stale signatures
PurgeNeutral 90 # Tokens with neutralish probabilities
PurgeUnused 90 # Unused tokens
PurgeHapaxes 30 # Tokens with less than 5 hits (hapaxes)
PurgeHits1S 15 # Tokens with only 1 spam hit
PurgeHits1I 15 # Tokens with only 1 innocent hit

LocalMX 127.0.0.1

SystemLog on
UserLog on

Opt out

ClamAVPort 3310

ClamAVHost 127.0.0.1
ClamAVResponse reject

ProcessorURLContext on
ProcessorBias on

/etc/clamd.conf:

```
##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração do ClamAV

LogFile /var/log/clamav/clamd.log
LogVerbose yes
PidFile /var/run/clamav/clamd.pid
TemporaryDirectory /var/lib/clamav/tmp
DatabaseDirectory /var/lib/clamav
LocalSocket /var/lib/clamav/clamd.socket
FixStaleSocket yes
TCPSocket 3310
TCPAddr 127.0.0.1
MaxThreads 64
ReadTimeout 300
FollowDirectorySymlinks yes
FollowFileSymlinks yes
User clamav
AllowSupplementaryGroups yes
ScanPE yes
ScanOLE2 yes
ScanMail yes
PhishingSignatures yes
ScanHTML yes
ScanArchive Yes
```

Apêndice E - Arquivo de configuração para os certificados auto-assinados

/etc/ssl/openssl.cnf:

```
##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo de configuração para a geração
##dos certificados auto-assinados

[ req ]
default_bits = 1024
encrypt_key = yes
distinguished_name = req_dn
x509_extensions = cert_type
prompt = no
[ req_dn ]
C=BR
```

```

ST=DF
L=Brasilia
O=Projeto final
OU=Projeto final
CN=servidor.projetofinal.org
emailAddress=postmaster@projetofinal.org
[ cert_type ]
nsCertType = Server

```

Apêndice F - Arquivos utilizados para automatizar a instalação do sistema

i586/auto_inst.cfg:

```

#!/usr/bin/perl -cw
#
# You should check the syntax of this file before using it in an auto-install.
# You can do this with 'perl -cw auto_inst.cfg.pl' or by executing this file
# (note the '#!/usr/bin/perl -cw' on the first line).
$o = {
    'printer' => {
        'configured' => {}
    },

    'libsane' => 0,

    'security_user' => "",

    'bootloader' => {
        'method' => 'grub-graphics',
        'crushMbr' => 1,
    },

    'useSupermount' => 0,

    'default_packages' => [
        'dhcp-client',
        'drakx-net-text',
        'grub',
        'iproute2',
        'iptables',
        'lynx',
        'rsync',
        'openssh-server',
        'pciutils',
        'urpmi',
        'vim-enhanced',
        'cdialog',
        'wget',
    ],

    'users' => [
        {
            'name' => 'vmail',
            'realname' => 'Mailbox user account',
            'shell' => '/dev/null',
        }
    ]
}

```

```

    'home' => '/home/vmail'
  },
  {
    'name' => 'projeto',
    'realname' => 'Usuário comum projeto',
    'password' => '123456',
    'shell' => '/bin/bash',
    'home' => '/home/projeto'
  },
],

'isUpgrade' => '0',

'locale' => {
  'country' => 'BR',
  'IM' => undef,
  'lang' => 'pt_BR',
  'langs' => {
    'pt_BR' => 1
  },
  'utf8' => '1'
},

'netc' => {
  'NETWORKING' => 'yes',
  'FORWARD_IPV4' => 'false',
  'HOSTNAME' => 'servidor.projetofinal.org',
  'DOMAINNAME' => 'projetofinal.org',
},

'net' => {
  'wireless' => {},
  'resolv' => {},
  'zeroconf' => {},
  'network' => {
    'NETWORKING' => 'yes',
    'HOSTNAME' => 'servidor.projetofinal.org'
  },
  'ifcfg' => {
    'eth0' => {
      'BOOTPROTO' => 'dhcp',
      'HOSTNAME' => 'servidor.projetofinal.org',
      'DOMAIN' => 'projetofinal.org',
      'ONBOOT' => 'yes',
      'DEVICE' => 'eth0',
      'MII_NOT_SUPPORTED' => 'yes'
    }
  },
  'PROFILE' => 'default'
},

'authentication' => {
  'shadow' => 1,
  'md5' => 1
},

'partitions' => [
  {
    'mntpoint' => '/boot',
    'type' => 'xfs',

```



```

    'size' => 256 << 11,
  },
  {
    'mntpoint' => 'pv00',
    'type' => '142',
    'size' => 256 << 11,
    'ratio' => 100,
  },
  {
    'VG_name' => 'vg00',
    'parts' => 'pv00'
  },
  {
    'mntpoint' => '/',
    'type' => 'xfs',
    'size' => 512 << 11,
    'hd' => 'vg00'
  },
  {
    'mntpoint' => '/tmp',
    'type' => 'xfs',
    'size' => 1024 << 11,
    'hd' => 'vg00'
  },
  {
    'mntpoint' => '/usr',
    'type' => 'xfs',
    'size' => 4096 << 11,
    'hd' => 'vg00'
  },
  {
    'mntpoint' => '/home',
    'type' => 'xfs',
    'size' => 4096 << 11,
    'hd' => 'vg00'
  },
  {
    'mntpoint' => '/var',
    'type' => 'xfs',
    'size' => 512 << 11,
    'hd' => 'vg00'
  },
  {
    'mntpoint' => 'swap',
    'type' => '130',
    'size' => 1024 << 11,
    'hd' => 'vg00'
  },
  {
    'mntpoint' => '/var/spool/deleted-maildirs',
    'type' => 'xfs',
    'size' => 256 << 11,
    'hd' => 'vg00'
  },
  {
    'mntpoint' => '/var/www',
    'type' => 'xfs',
    'size' => 256 << 11,
    'hd' => 'vg00'
  },
},

```

```

    {
      'mntpoint' => '/var/lib/pgsql',
      'type' => 'xfs',
      'size' => 512 << 11,
      'hd' => 'vg00'
    },
  ],

  'superuser' => {
    'realname' => 'root',
    'shell' => '/bin/bash',
    'password' => 'projeto123',
    'gid' => '0',
    'uid' => '0',
    'home' => '/root'
  },

  'services' => [
    'atd',
    'crond',
    'devfsd',
    'harddrake',
    'keytable',
    'kheader',
    'netfs',
    'network',
    'nfslock',
    'numlock',
    'partmon',
    'portmap',
    'random',
    'rawdevices',
    'syslog',
    'sshd',
  ],

  'mouse' => {
    'XMOUSETYPE' => 'ExplorerPS/2',
    'name' => 'Any PS/2 & USB mice',
    'EMULATEWHEEL' => undef,
    'device' => 'input/mice',
    'type' => 'Universal',
    'nbuttons' => 7,
    'MOUSETYPE' => 'ps/2',
    'wacom' => []
  },

  'autoExitInstall' => '0',

  'keyboard' => {
    'GRP_TOGGLE' => '',
    'KBCHARSET' => 'iso-8859-1',
    'KEYBOARD' => 'br',
    'KEYTABLE' => 'br-abnt2'
  },

  'manualFstab' => [],

  'timezone' => {
    'ntp' => 'pool.ntp.org',

```

```

    'timezone' => 'America/Sao_Paulo',
    'UTC' => 0
  },

  'X' => {
    'xdm' => 0
  },

  'partitioning' => {
    'auto_allocate' => '1',
    'clearall' => 1,
    'eraseBadPartitions' => 0
  },

  'security' => 2,

  'interactiveSteps' => [
    selectKeyboard,
  ],

  'postInstall' => "

cat <<EOF1 >> /etc/syslog.conf

*. *                                /dev/tty12
EOF1

mkdir /tmp/temp
mount -o loop /tmp/mdkinst.sqfs /tmp/temp
cp /tmp/temp/root/repo_conf.sh /root
rsync -av /tmp/temp/opt/configurador/ /opt/configurador/
umount /tmp/temp

ln -s /opt/configurador/bin/welcome.sh /usr/local/bin
ln -s /opt/configurador/bin/configurador.sh /usr/local/bin

mv /opt/configurador/bin/welcome.firstboot /etc/init.d/welcome
chkconfig --add welcome
"
};

```

i586/isolinux/isolinux.cfg:

```

default instalation
prompt 1
timeout 150
display help.msg
implicit 1
gfxboot bootlogo
label instalation
  kernel alt0/vmlinuz
  append initrd=alt0/all.rdz automatic=method:cdrom vga=788 splash=silent dir=/i586
auto_install=auto_inst.cfg mem=256m
label harddisk
  localboot 0x80
label linux
  kernel alt0/vmlinuz
  append initrd=alt0/all.rdz automatic=method:cdrom vga=788 splash=silent
label vgalo

```

```

kernel alt0/vmlinuz
append initrd=alt0/all.rdz automatic=method:cdrom vga=785
label vgahi
kernel alt0/vmlinuz
append initrd=alt0/all.rdz automatic=method:cdrom vga=791
label text
kernel alt0/vmlinuz
append initrd=alt0/all.rdz automatic=method:cdrom text
label rescue
kernel alt0/vmlinuz
append initrd=alt0/all.rdz automatic=method:cdrom rescue
label noacpi
kernel alt0/vmlinuz
append initrd=alt0/all.rdz automatic=method:cdrom vga=788 splash=silent acpi=off
label alt0
kernel alt0/vmlinuz
append initrd=alt0/all.rdz vga=788 splash=silent
label alt1
kernel alt1/vmlinuz
append initrd=alt1/all.rdz vga=788 splash=silent
label memtest
kernel memtest

```

Apêndice G - Arquivos para automatizar a configuração do sistema

/opt/configurador/bin/variables:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo com as variáveis necessárias para
##a execução do script configurador.sh

###Variáveis estáticas
##Não alterar!

CONF_DIR=/opt/configurador/confs
DMP_DIR=/opt/configurador/dmps
RPM_DIR=/opt/configurador/rpms
BIN_DIR=/opt/configurador/bin
MACHINE=$(uname -m)
SRV_NAME=$(grep HOSTNAME /etc/sysconfig/network | cut -d= -f2 | cut -d. -f1)

###Configuração do servidor

##Nome do domínio principal
DOM_NAME=projetofinal.sytes.net

##Habilitar os filtros SPF e Postgrey.
#Opções válidas:
#  Y - Habilita
#  N - Desabilita
FILTERS=N

###Configuração do certificado auto-assinado
##País (Sigla)

```

CERT_C=US

##Estado (Sigla)
CERT_ST=NY

##Localidade (cidade). Não retirar as aspas.
CERT_L="New York"

##Nome da empresa. Será inserido na tela de login do webmail. Não retirar as aspas.
CERT_O="Projeto final Sytes.net"

##Unidade Organizacional. Não retirar as aspas.
CERT_OU="Projeto final Sytes.net"

##E-mail para contato. Não alterar!
CERT_emailAddress=postmaster@\$DOM_NAME

/opt/configurador/bin/functions:

##Projeto Final
##Thiago de Almeida Milhomem
##Arquivo com as funções necessárias para
##a configuração do servidor.

#!/bin/bash

#Funções auxiliares

```
edita_opensslcnf (){
cat /etc/ssl/openssl.cnf | sed "s/C=.*C=$CERT_C/g" | sed "s/ST=.*ST=$CERT_ST/g" \
| sed "s/L=.*L=$CERT_L/g" | sed "s/OU=.*OU=$CERT_OU/g" \
| sed "s/O=.*O=$CERT_O/g" \
| sed "s/emailAddress=.*emailAddress=$CERT_emailAddress/g" > \
/tmp/openssl.cnf && mv /tmp/openssl.cnf /etc/ssl/openssl.cnf
}
```

```
cria_maildir_postmaster(){
mkdir /home/vmail/$DOM_NAME
chown vmail.vmail /home/vmail/$DOM_NAME
chmod 700 /home/vmail/$DOM_NAME
tar xzvf $CONF_DIR/postfixadmin/postmaster.initial.tar.gz -C /home/vmail/$DOM_NAME
}
```

```
cria_banco_postfix(){
service postgresql status | grep parado > /dev/null && service postgresql start
sleep 1
psql -U postgres -l | grep postfix > /dev/null
if [ $? = 0 ]; then
    service postgresql restart > /dev/null
    psql -U postgres -c "DROP DATABASE postfix;"
    psql -U postgres -c "DROP USER postfix;"
fi
```

```
psql -U postgres -c "CREATE USER postfix WITH PASSWORD '3ed4rf';"
psql -U postgres -c "CREATE DATABASE postfix OWNER postfix;"
sed "s/projetofinal.org/$DOM_NAME/g" $DMP_DIR/postfix.sql > /tmp/postfix.sql
psql -U postgres postfix < /tmp/postfix.sql
```

```

}

cria_banco_roundcubemail(){
service postgresql status | grep parado > /dev/null && service postgresql start
sleep 1
psql -U postgres -l | grep roundcubemail > /dev/null
if [ $? != 0 ]; then
    psql -U postgres -c "CREATE USER roundcube WITH PASSWORD '3ed4rf5tg';"
    psql -U postgres -c "CREATE DATABASE roundcubemail OWNER roundcube;"
    psql -U roundcube roundcubemail < $DMP_DIR/roundcubemail.sql
fi
}

cria_banco_dspam(){
service postgresql status | grep parado > /dev/null && service postgresql start
sleep 1
psql -U postgres -l | grep dspam > /dev/null
if [ $? != 0 ]; then
    psql -U postgres -c "CREATE USER dspam WITH PASSWORD '3ed4rf';"
    psql -U postgres -c "CREATE DATABASE dspam OWNER dspam;"
    psql -U postgres dspam < $DMP_DIR/dspam-users.sql
    psql -U postgres dspam < $DMP_DIR/dspam-objects.sql
    psql -U postgres dspam < $DMP_DIR/dspam-grant.sql
fi
}

```

#Funções principais

```

configura_certificados (){
    echo "Configurando Certificados.."
    sleep 1
    echo "-----"
    [ -d /etc/ssl/projeto ] && rm -rf /etc/ssl/projeto
    [ -f /etc/ssl/openssl.cnf ] && rm /etc/ssl/openssl.cnf
    cp $CONF_DIR/ssl/openssl.cnf /etc/ssl
    edita_opensslcnf
    cp $CONF_DIR/ssl/mkcert.sh /usr/local/bin
    chmod u+x /usr/local/bin/mkcert.sh
    /usr/local/bin/mkcert.sh postfixadmin admin.$DOM_NAME
    /usr/local/bin/mkcert.sh webmail webmail.$DOM_NAME
    /usr/local/bin/mkcert.sh email mail.$DOM_NAME
}

configura_dovecot (){
    echo "Configurando Dovecot.."
    sleep 1
    echo "-----"
    getent passwd dspam > /dev/null || useradd -r -d /var/lib/dspam dspam
    [ -f /etc/dovecot.conf ] && mv /etc/dovecot.conf /etc/dovecot.conf.old
    [ -f /etc/dovecot-pgsql.conf ] && rm /etc/dovecot-pgsql.conf
    cp $CONF_DIR/dovecot/* /etc

    service dovecot restart > /dev/null
    BOOT=$(chkconfig --list dovecot | awk '{ print $5 }' | cut -d: -f2)
    [ $BOOT != 'sim' ] && chkconfig dovecot on
}

configura_postfix(){

```

```

echo "Configurando Postfix.."
sleep 1
echo "-----"
[ -f /etc/postfix/main.cf ] && mv /etc/postfix/main.cf /etc/postfix/main.cf.old
[ -f /etc/postfix/master.cf ] && mv /etc/postfix/master.cf /etc/postfix/master.cf.old
[ -f /etc/postfix/transport ] && mv /etc/postfix/transport /etc/postfix/transport.old
[ -d /etc/postfix/pgsql ] || rsync -av $CONF_DIR/postfix/pgsql/ /etc/postfix/pgsql/

cp $CONF_DIR/postfix/main.cf /etc/postfix
cp $CONF_DIR/postfix/master.cf /etc/postfix
cp $CONF_DIR/postfix/transport /etc/postfix

sed "s/projetofinal.org/$DOM_NAME/g" /etc/postfix/main.cf > /tmp/main.cf && mv /tmp/main.cf
/etc/postfix/main.cf
sed "s/servidor/$SRV_NAME/g" /etc/postfix/main.cf > /tmp/main.cf && mv /tmp/main.cf
/etc/postfix/main.cf
sed "s/projetofinal.org/$DOM_NAME/g" /etc/postfix/transport > /tmp/transport && mv
/tmp/transport /etc/postfix/transport
cria_banco_postfix
BOOT=$(chkconfig --list postgresql | awk '{ print $5 }' | cut -d: -f2)
[ $BOOT != 'sim' ] && chkconfig postgresql on

service postfix restart > /dev/null
BOOT=$(chkconfig --list postfix | awk '{ print $5 }' | cut -d: -f2)
[ $BOOT != 'sim' ] && chkconfig postfix on
}

configura_postfixadmin(){
echo "Configurando PostfixAdmin.."
sleep 1
echo "-----"
[ -d /var/www/postfixadmin ] || tar xzvf
$CONF_DIR/postfixadmin/postfixadmin_2.3rc2_noarch.tar.gz -C /var/www && chown -R
apache.apache /var/www/postfixadmin
[ -f /var/www/postfixadmin/config.inc.php ] && rm /var/www/postfixadmin/config.inc.php
cp $CONF_DIR/postfixadmin/config.inc.php /var/www/postfixadmin
[ -d /home/vmail/$DOM_NAME/postmaster ] || cria_maildir_postmaster
[ -f /usr/local/bin/postfixadmin-domain-postdeletion.sh ] || cp
$CONF_DIR/postfixadmin/postfixadmin-domain-postdeletion.sh /usr/local/bin
[ -f /usr/local/bin/postfixadmin-mailbox-postdeletion.sh ] || cp
$CONF_DIR/postfixadmin/postfixadmin-mailbox-postdeletion.sh /usr/local/bin
chmod u+x /usr/local/bin/*
[ -f /etc/sudoers ] && mv /etc/sudoers /etc/sudoers.old
cp $CONF_DIR/postfixadmin/sudoers /etc
[ -f /etc/httpd/conf/vhosts.d/00_default_vhosts.conf ] && rm
/etc/httpd/conf/vhosts.d/00_default_vhosts.conf
[ -f /etc/httpd/conf/vhosts.d/01_default_ssl_vhost.conf ] && rm
/etc/httpd/conf/vhosts.d/01_default_ssl_vhost.conf
[ -f /etc/httpd/conf/vhosts.d/postfixadmin.conf ] && rm
/etc/httpd/conf/vhosts.d/postfixadmin.conf
[ -f /etc/httpd/conf/vhosts.d/postfixadmin-ssl.conf ] && rm /etc/httpd/conf/vhosts.d/postfixadmin-
ssl.conf
cp $CONF_DIR/apache/postfixadmin.conf /etc/httpd/conf/vhosts.d/
cp $CONF_DIR/apache/postfixadmin-ssl.conf /etc/httpd/conf/vhosts.d/

sed "s/projetofinal.org/$DOM_NAME/g" /etc/httpd/conf/vhosts.d/postfixadmin.conf >
/tmp/postfixadmin.conf && mv /tmp/postfixadmin.conf /etc/httpd/conf/vhosts.d/postfixadmin.conf

```

```

        sed "s/projetofinal.org/$DOM_NAME/g" /etc/httpd/conf/vhosts.d/postfixadmin-ssl.conf >
/tmp/postfixadmin-ssl.conf && mv /tmp/postfixadmin-ssl.conf /etc/httpd/conf/vhosts.d/postfixadmin-
ssl.conf
        sed "s/projetofinal.org/$DOM_NAME/g" /var/www/postfixadmin/config.inc.php >
/tmp/config.inc.php && mv /tmp/config.inc.php /var/www/postfixadmin/config.inc.php

        service httpd restart > /dev/null
        BOOT=$(chkconfig --list httpd | awk '{ print $5 }' | cut -d: -f2)
        [ $BOOT != 'sim' ] && chkconfig httpd on
    }

configura_webmail(){
    echo "Configurando webmail.."
    sleep 1
    echo "-----"
    [ -d /var/www/roundcubemail ] || tar xzvf
$CONF_DIR/roundcube/roundcubemail_0.2.1_noarch.tar.gz -C /var/www && chown -R
apache.apache /var/www/roundcubemail
    [ -f /var/www/roundcubemail/config/main.inc.php ] && rm
/var/www/roundcubemail/config/main.inc.php
    cp $CONF_DIR/roundcube/main.inc.php /var/www/roundcubemail/config/

    [ -f /etc/httpd/conf/vhosts.d/roundcubemail.conf ] && rm
/etc/httpd/conf/vhosts.d/roundcubemail.conf
    [ -f /etc/httpd/conf/vhosts.d/roundcubemail-ssl.conf ] && rm
/etc/httpd/conf/vhosts.d/roundcubemail-ssl.conf
    cp $CONF_DIR/apache/roundcubemail.conf /etc/httpd/conf/vhosts.d/
    cp $CONF_DIR/apache/roundcubemail-ssl.conf /etc/httpd/conf/vhosts.d/

    sed "s/projetofinal.org/$DOM_NAME/g" /etc/httpd/conf/vhosts.d/roundcubemail.conf >
/tmp/roundcubemail.conf && mv /tmp/roundcubemail.conf /etc/httpd/conf/vhosts.d/roundcubemail.conf
    sed "s/projetofinal.org/$DOM_NAME/g" /etc/httpd/conf/vhosts.d/roundcubemail-ssl.conf >
/tmp/roundcubemail-ssl.conf && mv /tmp/roundcubemail-ssl.conf
/etc/httpd/conf/vhosts.d/roundcubemail-ssl.conf
    sed "s/projetofinal.org/$DOM_NAME/g" /var/www/roundcubemail/config/main.inc.php >
/tmp/main.inc.php && mv /tmp/main.inc.php /var/www/roundcubemail/config/main.inc.php
    sed "s/Projeto Final/$CERT_O/g" /var/www/roundcubemail/config/main.inc.php >
/tmp/main.inc.php && mv /tmp/main.inc.php /var/www/roundcubemail/config/main.inc.php

    cria_banco_roundcubemail
    service httpd restart > /dev/null
}

configura_filtros(){
    echo "Configurando filtros.."
    sleep 1
    echo "-----"
    if [ $MACHINE != x86_64 ];then
        [ -f /usr/lib/postfix/postfix-policyd-spf-perl ] || cp $CONF_DIR/postfix/postfix-policyd-spf-
perl /usr/lib/postfix
    else
        [ -f /usr/lib64/postfix/postfix-policyd-spf-perl ] || cp $CONF_DIR/postfix/postfix-policyd-
spf-perl /usr/lib64/postfix
    fi

    [ -f /etc/sysconfig/postgrey ] && mv /etc/sysconfig/postgrey /etc/sysconfig/postgrey.old
    cp $CONF_DIR/postfix/postgrey /etc/sysconfig
    service postgrey restart > /dev/null
}

```



```

    if [ $FILTERS != N ];then
        ( cat /etc/postfix/main.cf | grep inet:127.0.0.1:10031, | grep '#' > /dev/null)
        if [ $? = 0 ];then
            sed 's/#check_policy_service inet:127.0.0.1:10031,/check_policy_service
inet:127.0.0.1:10031,/g' /etc/postfix/main.cf > /tmp/main.cf
            mv /tmp/main.cf /etc/postfix/main.cf
        fi

        ( cat /etc/postfix/main.cf | grep unix:private/policy, | grep '#' > /dev/null)
        if [ $? = 0 ];then
            sed 's/#check_policy_service unix:private\policy,/check_policy_service
unix:private\policy,/g' /etc/postfix/main.cf > /tmp/main.cf
            mv /tmp/main.cf /etc/postfix/main.cf
        fi
    else
        ( cat /etc/postfix/main.cf | grep inet:127.0.0.1:10031, | grep '#' > /dev/null)
        if [ $? = 1 ];then
            sed 's/check_policy_service inet:127.0.0.1:10031,/check_policy_service
inet:127.0.0.1:10031,/g' /etc/postfix/main.cf > /tmp/main.cf
            mv /tmp/main.cf /etc/postfix/main.cf
        fi

        ( cat /etc/postfix/main.cf | grep unix:private/policy, | grep '#' > /dev/null)
        if [ $? = 1 ];then
            sed 's/check_policy_service unix:private\policy,/check_policy_service
unix:private\policy,/g' /etc/postfix/main.cf > /tmp/main.cf
            mv /tmp/main.cf /etc/postfix/main.cf
        fi
    fi

    service postfix restart > /dev/null
}

configura_antispam(){
    echo "Configurando antispam.."
    sleep 1
    echo "-----"
    [ -f /etc/dspam.conf ] && mv /etc/dspam.conf /etc/dspam.conf.old
    cp $CONF_DIR/dspam/dspam.conf /etc
    if [ $MACHINE != x86_64 ];then
        chown vmail.vmail /usr/lib/dovecot/deliver
        chmod ug+s /usr/lib/dovecot/deliver
    else
        chown vmail.vmail /usr/lib64/dovecot/deliver
        chmod ug+s /usr/lib64/dovecot/deliver
    fi

    [ -f /etc/postfix/dspam_filter_access ] || cp $CONF_DIR/postfix/dspam_filter_access
/etc/postfix

    [ -f /etc/clamd.conf ] && mv /etc/clamd.conf /etc/clamd.conf.old
    cp $CONF_DIR/dspam/clamd.conf /etc
    service clamd restart > /dev/null

    [ -f /var/lib/dspam/default.prefs ] || cp $CONF_DIR/dspam/default.prefs /var/lib/dspam
    [ -f /etc/httpd/conf/webapps.d/dspam.conf ] && mv /etc/httpd/conf/webapps.d/dspam.conf
/etc/httpd/conf/webapps.d/dspam.conf.old
    [ -f /usr/share/dspam/cgi-bin/configure.pl ] && mv /usr/share/dspam/cgi-bin/configure.pl
/usr/share/dspam/cgi-bin/configure.pl.old

```

```

cp $CONF_DIR/dspam/configure.pl /usr/share/dspam/cgi-bin
[ -f /usr/share/dspam/cgi-bin/admins ] && mv /usr/share/dspam/cgi-bin/admins
/usr/share/dspam/cgi-bin/admins.old
sed "s/projetofinal.org/$DOM_NAME/g" $CONF_DIR/dspam/admins > /usr/share/dspam/cgi-
bin/admins

( cat /etc/postfix/main.cf | grep 'pcre:/etc/postfix/dspam_filter_access' | grep '#' > /dev/null )
if [ $? = 0 ];then
    sed 's/#check_client_access\
pcre:Vetc\postfix\dspam_filter_access/check_client_access\
pcre:Vetc\postfix\dspam_filter_access/g' /etc/postfix/main.cf > /tmp/main.cf
    mv /tmp/main.cf /etc/postfix/main.cf
    service postfix restart > /dev/null
fi

cria_banco_dspam

usermod -aG mail apache
chmod -R 770 /var/lib/dspam
chmod -R g+s /var/lib/dspam
chown -R dspam:mail /var/lib/dspam

service httpd restart > /dev/null
}

```

/opt/configurador/bin/configurador.sh:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Script principal
##Configura o servidor a partir das funções
##localizadas em functions

#!/bin/bash

if [ $USER != 'root' ];then
    echo "Execute esse script como root"
    exit 1
fi

if [ $# -ne 1 ];then
    echo "Modo de uso: configurador.sh (certificados|servidor|filtros|antispam|full)"
    exit 1
fi

. /opt/configurador/bin/variables

source /opt/configurador/bin/functions

case $1 in
    certificados)
        configura_certificados
        ;;
    servidor)
        configura_dovecot
        configura_postfix
        configura_postfixadmin

```

```

        configura_webmail
        ;;
    filtros)
        configura_filtros
        ;;
    antispam)
        configura_antispam
        ;;
    full)
        configura_certificados
        configura_dovecot
        configura_postfix
        configura_postfixadmin
        configura_webmail
        configura_filtros
        configura_antispam
        ;;
    *)
        echo "Modo de uso: configurador.sh (certificados|servidor|filtros|antispam|full)"
        exit 1
        ;;
esac

```

/opt/configurador/bin/bemvindo.sh:

```

##Projeto Final
##Thiago de Almeida Milhomem
##Script auxiliar
##Realiza a configuração inicial do servidor.

```

```

#Carrega configurações de terminal
./root/.bashrc

```

```

#Variáveis iniciais
BASEDIR=/opt/configurador

```

```

BACKTITLE="Projeto Final - Script de configuração inicial - Thiago de Almeida Milhomem"
SCREEN1=$(cat $BASEDIR/bin/textos/tela_inicial)
SCREEN2=$(cat $BASEDIR/bin/textos/instalacao_pacotes)
SCREEN3=$(cat $BASEDIR/bin/textos/alteracao_senha_root)
SCREEN4=$(cat $BASEDIR/bin/textos/alteracao_senha_projeto)
SCREEN5=$(cat $BASEDIR/bin/textos/configuracao_ip)
SCREEN6=$(cat $BASEDIR/bin/textos/tela_final)

```

#Funções

```

verifica_retorno(){
    retorno=$?
    [ $retorno -eq 0 ] && proxima=$anterior
    [ $retorno -eq 255 ] && break
}

```

```

tela_inicial(){
    dialog --backtitle "$BACKTITLE" --title "Bem-vindo" --yes-label "Sair" --no-label "OK" --yesno
"$SCREEN1" 18 60
}

```

```

instalacao_pacotes(){
    dialog --backtitle "$BACKTITLE" --title "Instalação de pacotes" --yes-label "Voltar" --no-label
"Avançar" --yesno "$SCREEN2" 16 60
}

mensagem_pacotes_instalados(){
    dialog --backtitle "$BACKTITLE" --title "Instalação de pacotes" --infobox "Todos os pacotes
necessários já estão instalados." 7 33
    sleep 2
}

alteracao_senha_root(){
    dialog --backtitle "$BACKTITLE" --title "Alteração de senhas" --yes-label "Voltar" --no-label
"Avançar" --yesno "$SCREEN3" 17 60
}

alteracao_senha_projeto(){
    dialog --backtitle "$BACKTITLE" --title "Alteração de senhas" --yes-label "Voltar" --no-label
"Avançar" --yesno "$SCREEN4" 20 65
}

configuracao_ip(){
    dialog --backtitle "$BACKTITLE" --title "Configuração de rede" --yes-label "Voltar" --no-label
"Avançar" --yesno "$SCREEN5" 20 71
}

tela_final(){
    dialog --backtitle "$BACKTITLE" --title "Bem-vindo" --yes-label "Voltar" --no-label "Finalizar" --
yesno "$SCREEN6" 20 75
}

#Loop de menu
proxima=inicial

while : ; do

    case "$proxima" in
        inicial)
            proxima=pacotes
            tela_inicial
            retorno=$?
            [ $retorno -eq 0 ] && break
            [ $retorno -eq 255 ] && break
            ;;
        pacotes)
            anterior=inicial
            proxima=senha_root
            instalacao_pacotes
            verifica_retorno
            if [ $retorno -eq 1 ];then
                if [ -f $BASEDIR/rpms/.lock ];then
                    rpm -Uvh $BASEDIR/rpms/*
                    rm $BASEDIR/rpms/.lock
                else
                    mensagem_pacotes_instalados
                fi
            fi
        fi
    esac
done

```

```

;;
senha_root)
    anterior=pacotes
    proxima=senha_projeto
    alteracao_senha_root
    verifica_retorno
    [ $retorno -eq 1 ] && passwd root
;;
senha_projeto)
    anterior=senha_root
    proxima=ip
    alteracao_senha_projeto
    verifica_retorno
    [ $retorno -eq 1 ] && passwd projeto
;;
ip)
    anterior=senha_projeto
    proxima=final
    configuracao_ip
    verifica_retorno
    [ $retorno -eq 1 ] && drakconnect
;;
final)
    anterior=ip
    tela_final
    retorno=$?
    [ $retorno = 1 ] && break
    [ $retorno -eq 255 ] && break
    [ $retorno = 0 ] && proxima=$anterior
;;
*)
    echo "Janela desconhecida '$proxima'."
    exit 1
esac

done
clear

```

/opt/configurador/bin/textos/tela_inicial:

Bem-vindo!

Esse script irá auxiliar a configuração inicial do servidor.

Para interromper esse script, pressione a tecla ESC a qualquer momento.

Caso deseje executar esse script novamente, digite o comando 'welcome.sh' no terminal.

/opt/configurador/bin/textos/instalacao_pacotes:

Para o funcionamento correto do servidor, alguns pacotes precisam ser instalados.

Selecione 'Avançar' para continuar.

/opt/configurador/bin/textos/alteração_senha_root:

O usuário root possui privilégios administrativos no sistema. Através dele todas as configurações do servidor são realizadas.

Será solicitada a nova senha do root após essa mensagem.

Selecione 'Avançar' para continuar.

/opt/configurador/bin/textos/alteração_senha_projeto:

O usuário comum projeto não possui privilégios administrativos no sistema.

Esse usuário pode ser utilizado para realizar acesso remoto via SSH no servidor, já que o acesso com o usuário root está desabilitado, por questões de segurança.

Será solicitada a nova senha do usuário projeto após essa mensagem.

Selecione 'Avançar' para continuar.

/opt/configurador/bin/textos/configuração_rede:

Nesta etapa serão realizadas as configurações de rede do servidor.

Uma ferramenta para configuração de rede será executada após essa mensagem.

O endereço IP configurado deverá possuir liberação no firewall (caso exista) para alguns serviços. Para maiores informações, verifique o arquivo Leia-me.txt situado em /opt/configurador.

Selecione 'Avançar' para continuar.

/opt/configurador/bin/textos/tela_final:

Parabéns!

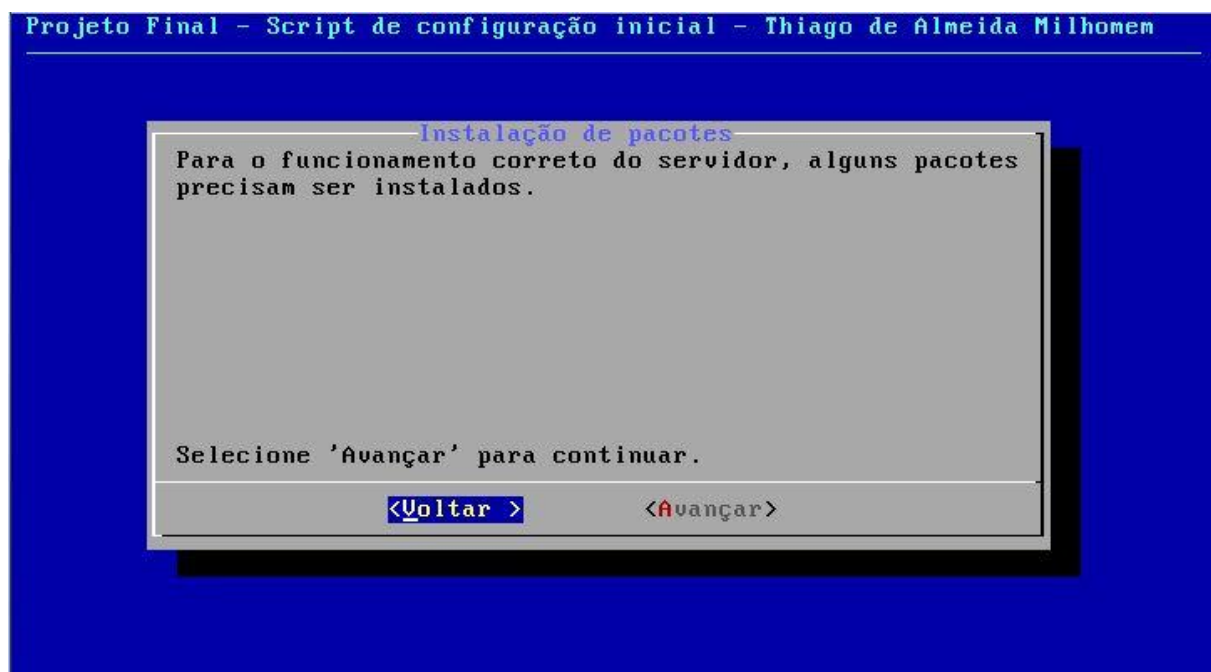
A configuração inicial do servidor foi finalizada.

No entanto, ainda restam algumas configurações para que o servidor esteja pronto.

Todas as informações necessárias realizar as configurações restantes do servidor se encontram no arquivo Leia-me.txt, situado no diretório /opt/configurador.

Selecione 'Finalizar' para terminar este script.

Apêndice H - Telas do script bemvindo.sh



Projeto Final - Script de configuração inicial - Thiago de Almeida Milhomen

Alteração de senhas

O usuário root possui privilégios administrativos no sistema. Através dele todas as configurações do servidor são realizadas.

Será solicitada a nova senha do root após essa mensagem.

Selecione 'Avançar' para continuar.

<V^oltar >

<A^vançar>

Projeto Final - Script de configuração inicial - Thiago de Almeida Milhomen

Alteração de senhas

O usuário comum projeto não possui privilégios administrativos no sistema.

Esse usuário pode ser utilizado para realizar acesso remoto via SSH no servidor, já que o acesso com o usuário root está desabilitado, por questões de segurança.

Será solicitada a nova senha do usuário projeto após essa mensagem.

Selecione 'Avançar' para continuar.

<V^oltar >

<A^vançar>

Projeto Final - Script de configuração inicial - Thiago de Almeida Milhomem

Configuração de rede

Nesta etapa serão realizadas as configurações de rede do servidor.

Uma ferramenta para configuração de rede será acionada após essa mensagem.

O endereço IP configurado deverá possuir liberação no firewall (caso exista) para alguns serviços.

Para maiores informações, verifique o arquivo Leia-me.txt situado em /opt/configurador.

Selecione 'Avançar' para continuar.

<Voltar >

<Avançar>

Projeto Final - Script de configuração inicial - Thiago de Almeida Milhomem

Ben-vindo

Parabéns!

A configuração inicial do servidor foi finalizada.

No entanto, ainda restam algumas configurações para que o servidor esteja pronto.

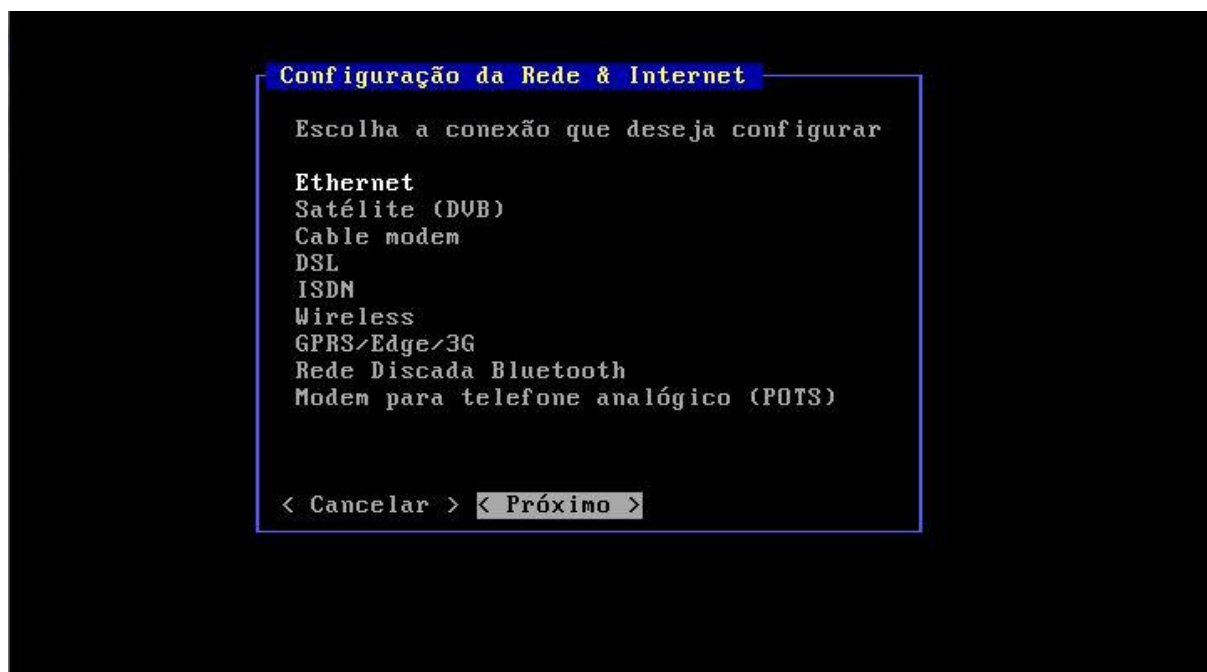
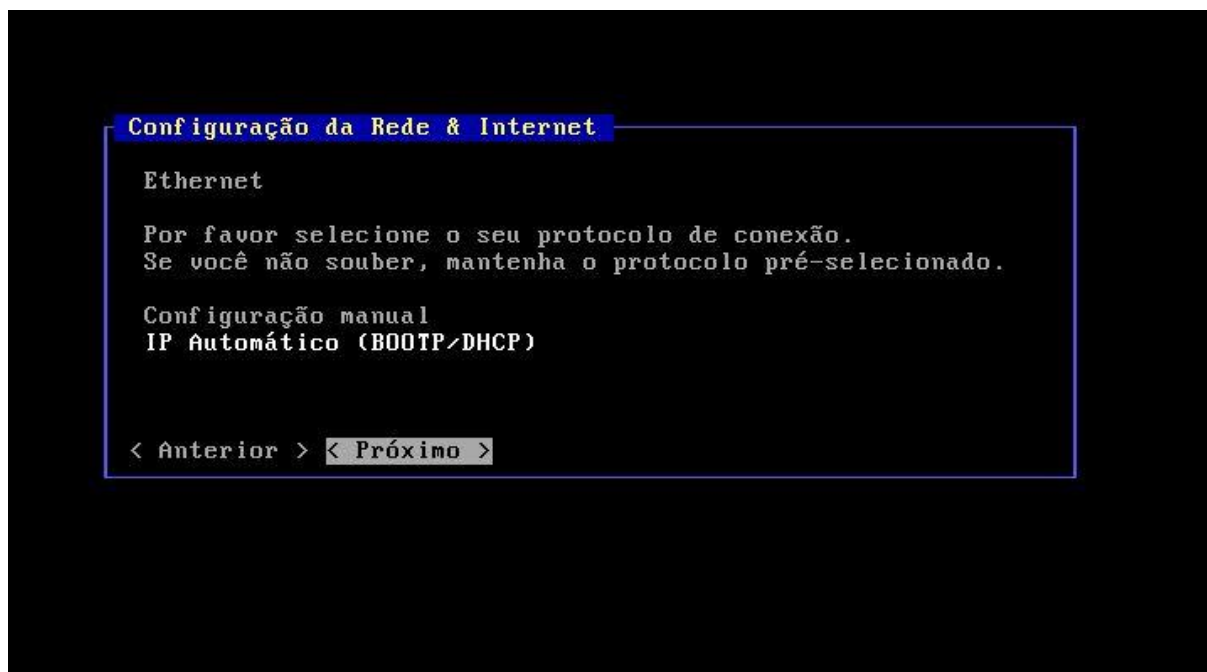
Todas as informações necessárias realizar as configurações restantes do servidor se encontram no arquivo Leia-me.txt, situado no diretório /opt/configurador.

Selecione 'Finalizar' para terminar este script.

< Voltar >

<Finalizar>

Anexo A - Utilitário de configuração de rede drakconnect



Configuração da Rede & Internet

Ethernet

Configurações de IP

☒ Obter servidores DNS do DHCPServidor DNS 1 ☐Servidor DNS 2 ☐☐ Obter o nome da máquina a partir do endereço DHCPNome da máquina ☐

<+> Avançado

< Anterior > < Próximo >

Configuração da Rede & Internet

Ethernet

Controle de conexão

☐ Permitir que usuários gerenciem a conexão☒ Iniciar a conexão no boot

<+> Avançado

< Anterior > < Próximo >